Conteúdo de Cripto 2021.2

Hugo Nobrega

Sumário

Resumo Geral

Aula 0
Parte 1: Introdução à disciplina e questões administrativas; Parte 2: Exemplos motivadores (cifra de César e cifra multiplicativa)

Aula 01
Parte 1: Inversos multiplicativos no mundo circular; Parte 2: Abstração em computação e matemática; expressões e variáveis em python

Aula 02
Parte 1: Definindo funções em Python; Parte 2: Erros; tipos numéricos

Aula 03
Parte 1: Definição de variáveis, atribuições simultâneas, escopo, alias; Parte 2: Tipos numéricos, booleanos

Aula 04
Parte 1: Dados Booleanos (True/False); comandos condicionais (if/elif/else); Parte 2: condicional (while) e sequencial (for)

Aula 05
Parte 1: Recap de while/for; strings; indexação, fatiamento; Parte 2: Mais sobre strings; métodos de strings; cifra de César (início)

Aula 06
Parte 1: Cifra de César (implementação); Parte 2: Tuplas e listas; mutabilidade

Aula 07
Parte 1: Mais sobre mutabilidade; hasheabilidade; Parte 2: Dicionários

Aula 08
Parte 1: Algoritmos, teoremas e definições em matemática; Parte 2: Provas em matemática; exemplo de uma prova

Aula 09
Parte 1: Dúvidas; Parte 2: Recíproca de um teorema; contrapositiva; Parte 3: Divisibilidade (definição e algumas propriedades)

Aula 10
Parte 1: Mais sobre divisibilidade; mdc (definição); Parte 2: Algumas propriedades do mdc; algoritmo ingênuo do mdc; algoritmo de Euclides (introdução)

Aula 11
Parte 1: Algoritmo de Euclides (enunciado completo e implementação); Parte 2: Terminação do algoritmo de Euclides; corretude (ideia)

Aula 12
Parte 1: Dúvidas da lista 1; Parte 2: Corretude do Algoritmo de Euclides (ideia novamente)

Aula 13
Parte 1: Corretude do Algoritmo de Euclides; Parte 2: Lema para a prova da corretude

Aula 14
Parte 1: Recap do conteúdo; Parte 2: Prova do Teorema de Bézout; algoritmo de Euclides estendido (início)

Aula 15
Parte 1: Enunciado do Algoritmo de Euclides Estendido (AEE); Parte 2: Implementação do AEE; Parte 3: Terminação e Corretude do AEE

Aula 16

Parte 1: Estendendo o AEE para números negativos; Parte 2: Números primos e o Teorema Fundamental da Aritmética (TFA, enunciado); Parte 3: Prova de existência no TFA; algoritmo de fatoração em primos (enunciado e implementação)

Aula 17

Parte 1: Comentários sobre a Lista 2 e sobre as regras de colaboração entre alunos; Parte 2: terminação do algoritmo de fatoração em primos; Parte 3: corretude do algoritmo de fatoração em primos

Aula 18

Parte 1: Resolução do exercício L2Q1c; Parte 2: Otimizando a fatoração: o menor fator de um composto é menor ou igual à sua raiz; Parte 3: Comentários sobre a dificuldade do problema da fatoração; Parte 4: Unicidade no TFA (ideia); a Propriedade Fundamental dos Primos (enunciado)

Aula 19

Parte 1: Propriedade Fundamental dos Primos (prova); Parte 2: Unicidade no TFA (prova)

Aula 20

Parte 1: Comentários sobre a Lista 02; Parte 2: Infinitude dos Primos (enunciado); Parte 3: Infinitude dos Primos (prova); Parte 4: Definições recursivas (motivação)

Aula 21

Parte 1: Dúvidas da Aula 20, definições recursivas (exemplos e discussão); Parte 2: Definições recursivas (definição oficial); Parte 3: Implementando definições recursivas em programação

Aula 22

Parte 1: Comentários sobre a Lista 2; Parte 2: Torres de Hanói (definição); Parte 3: Torres de Hanói (algoritmo recursivo); Parte 4: Torres de Hanói (implementação em Python)

Aula 23

Parte 1: Dúvidas da Lista 3; Parte 2: O método de prova por indução; exemplo (soma dos naturais até n); Parte 3: Outro exemplo de prova por indução (fórmula de Binet para Fibonacci)

Aula 24

Parte 1: Recapitulação de indução; Parte 2: Prova de terminação do algoritmo de Hanói; Parte 3: Prova de corretude do algoritmo de Hanói; quantidade de movimentos feitos pelo algoritmo

Aula 25

Parte 1: Dúvidas da Lista 3; Parte 2: Número de passos do algoritmo de Hanói; Parte 3: Relações; reflexividade, simetria, transitividade, relações de equivalência

Aula 26

Parte 1: Recapitulação de relações; a "cara" das relações de equivalência; Parte 2: Quociente por relação de equivalência; inteiros módulo n; aritmética modular (introdução)

Aula 27

Parte 1: Aritmética Modular (adição, subtração, multiplicação); Parte 2: Aritmética Modular (divisão); Parte 3: Aritmética Modular (não há potenciação)

Aula 28

Parte 1: Revisão de recursão e indução; Aritmética Modular (elevando uma classe de equivalência a um número natural); exemplos; Parte 2: Mais exemplos de exponenciação modular; Parte 3: Pequeno Teorema de Fermat (motivação e enunciado)

Aula 29

Parte 1: Dúvidas da Lista 4; Parte 2: Equivalência entre PTF1 e PTF2; Parte 3: Prova do PTF1

Aula 30

Parte 1: Aplicação do PTF: cálculo de potências em aritmética modular; Parte 2: Aplicação do PTF: teste de primalidade; Parte 3: Pseudoprimos de Fermat, números de Carmichael

Aula 31

Parte 1: Organização do final da disciplina; Parte 2: Recapitulação de resultados antigos usando a linguagem da aritmética modular; Parte 3: O teste de primalidade de Miller–Rabin

Aula 32

Parte 1: Anúncios sobre o trabalho final e monitoria; Parte 2: Implementação e discussão do Teste de Miller–Rabin; Parte 3: A ideia do RSA

Aula 33

Parte 1: Como garantir b**(ed) = b mod pq; Parte 2: RSA; Parte 3: A conversão de textos para números; a escolha dos primos p, q

Hugo Nóbrega

Python 3 → minicurso nas primeiras aulas

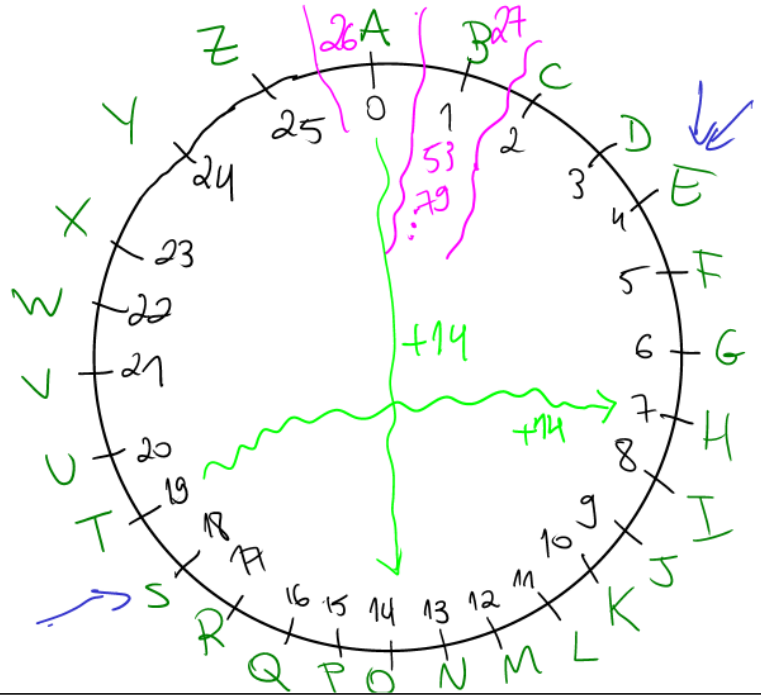
Questão E1

Frequências

S → 15%
C → 12%
O → 12%

a? k? rotação foi de 18 posições

e? o a? rotação de 14



Para desfazer uma rotação de 14 no sentido horário, posso

- rodar 14 no sentido antihorário
- rodar 26-14 = 12 no sentido horário

Questão E2

Frequências

0 a → 15% ← a? e? o?
12 m → 11% ← a? e? o?
16 q → 10% ← a? e? o?
13 n → 8,5%

por tentativa e erro, podemos ver que não foi encriptado por soma

Supondo que a enciptação foi feita por multiplicação, digamos multiplicação por e, a desciptação seria "dividir por e", ou multiplicar por d, tal que "$e \cdot d = 1$"

(d inteiro)

isso aqui é querer demais!

na verdade, basta que $e \cdot d$ caia na mesma casa que o 1 (ou seja, que seja da forma $1 + 26 \cdot n$ com n inteiro)

---

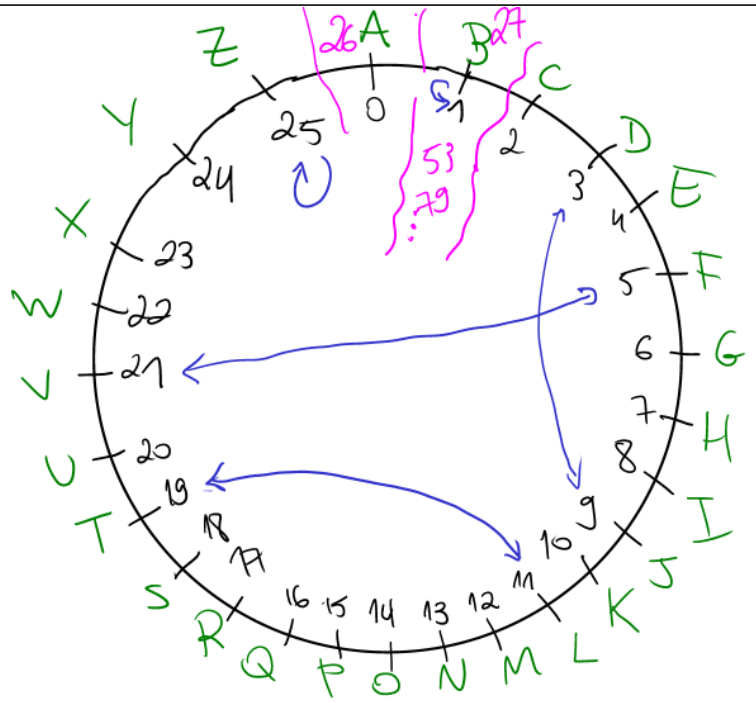
se a desciptação foi multiplicar por $d = 9$ então enciptar foi "dividir por 9", ou seja, multiplicar por $e = 3$

pois $e \cdot d = 27$, que está na casa do 1 no mundo cíclico de tamanho 26.

Outros exemplos de "inversos multiplicativos" no mundo circular de tamanho 26

e	inverso
3	9
1	27 ou 1
9	3
5	21
2	não tem

⇝ queremos d tal que 5.d caia na casa do 1



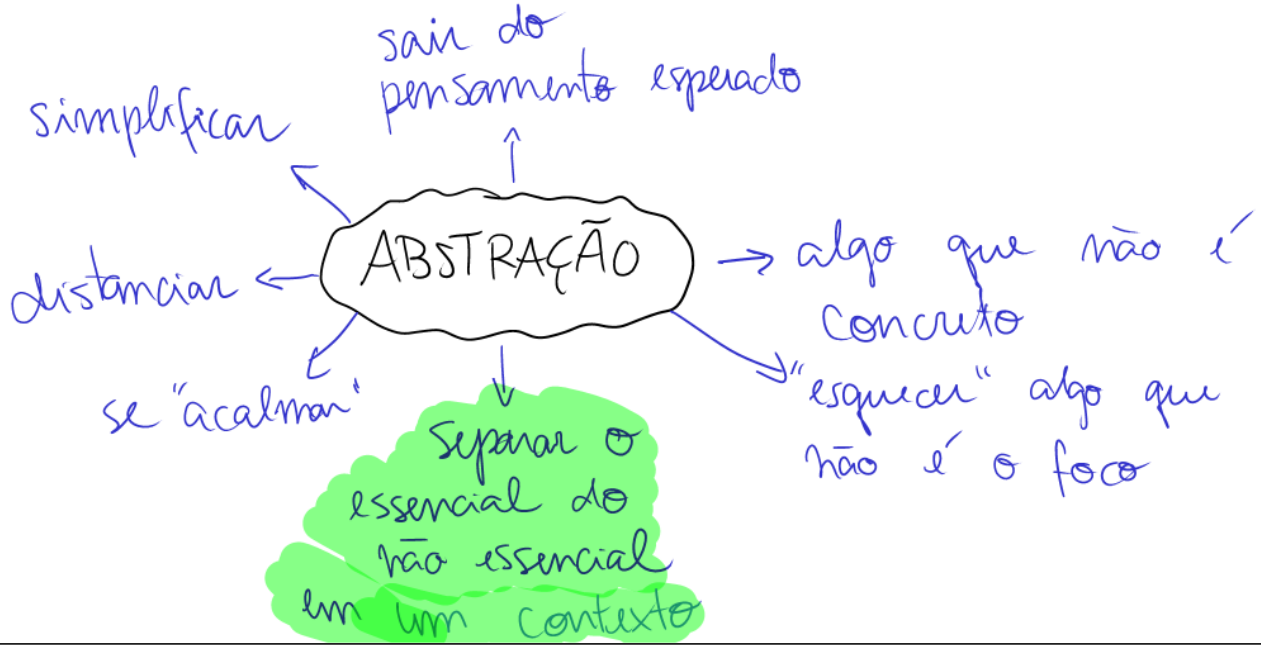
e	inverso
8	não tem
13	não tem
39	não tem
19	11 ⇝ pois $11 \cdot 19 = 1 + 26 \cdot 8$

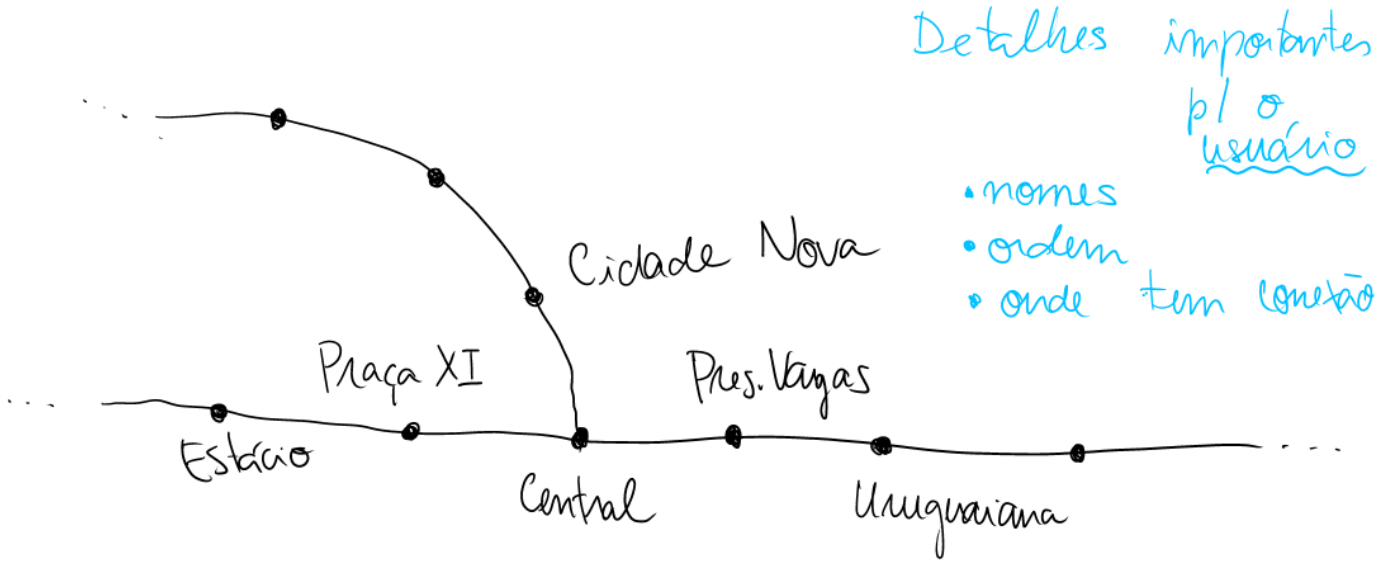
(não existem d, n inteiros tais que $13 \cdot d = 1 + 26 \cdot n$)

Moral da história (conjectura): os inversíveis no mundo circular de tamanho 26 são os números ímpares que não são múltiplos de 13

[ou seja, número cujo mdc com 26 é 1]

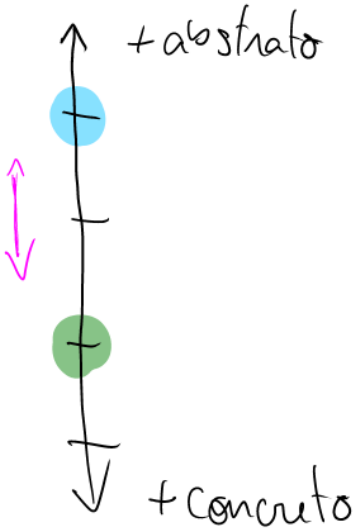
o inverso de 25 : podemos "trocar" 25 por -1 (caem na mesma casa!) e procurar d tal que d.(-1) caia na casa do 1. Podemos pegar d=-1! ou d=25





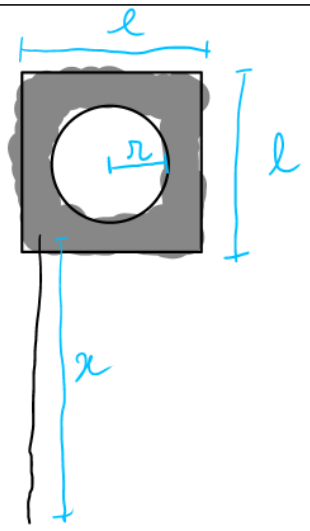
Os detalhes importantes para quem constrói o metrô são outros

- composição do solo
- distâncias
- etc



Ao programar, uma pessoa é construtora do programa, mas é também usuária de programas já escritos (por ela mesma ou outros!)

Na matemática é parecido: construímos provas de teoremas usando teoremas já provados anteriormente



Custo: área pintada + $2 \cdot x$

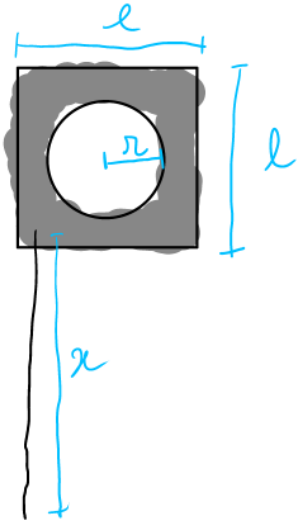
$= \ell^2 - \pi \cdot r^2 + 2 \cdot x$

Exemplo

$\ell = 3$

$r = 1$

$x = 4$



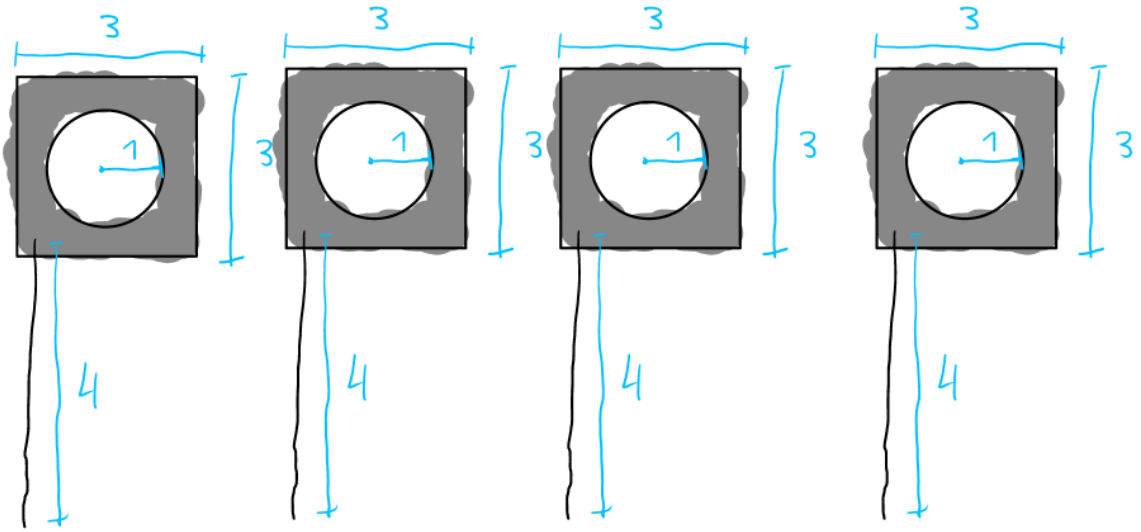
Dúvidas / Comentários

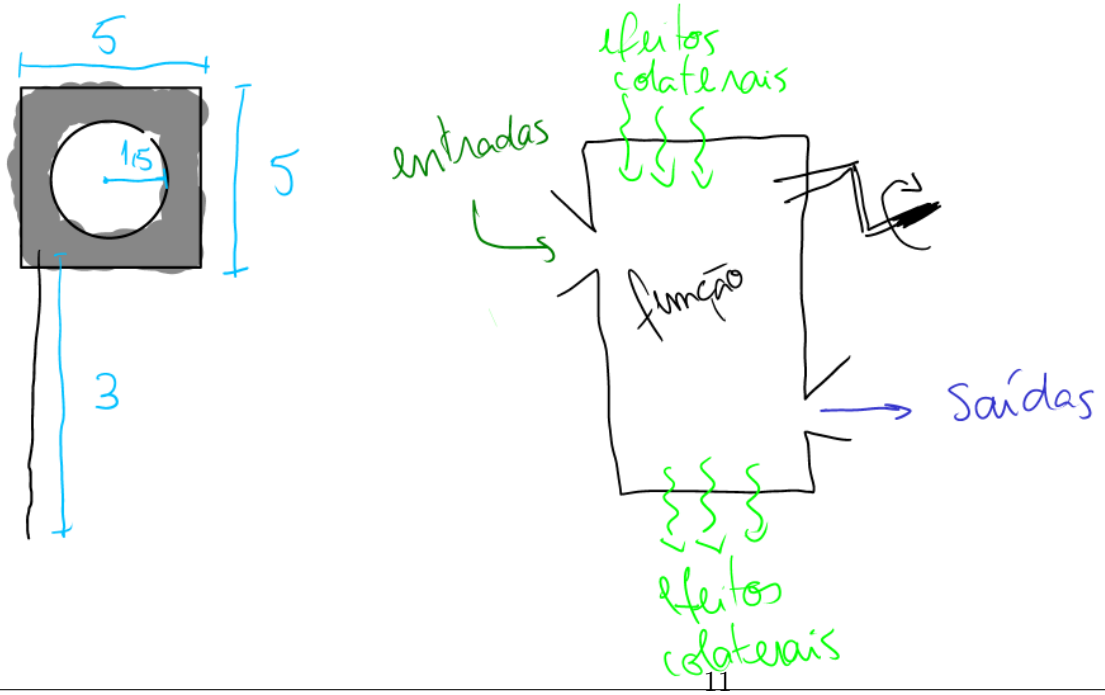
Vantagens de usar variáveis

→ evitar "retrabalho" ← computador ← humano

→ organização ← humano

→ ...





Dúvidas / comentários?

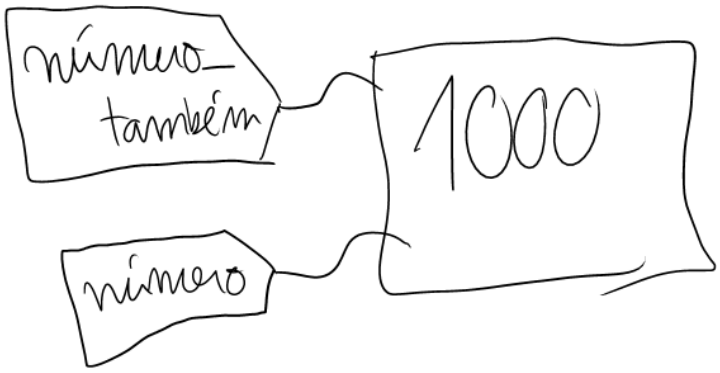
<nome> = <expressão>

↳ avaliada, gerando um valor

↳ <nome> vira uma nova expressão válida, com o valor que acabamos de calcular

Na vida real, = serve para atribuir valor ("seja $x = 10$")
MATEMÁTICA
e para comparar valores ("encontre x tal que $2 \cdot x = 100$")

Em Python, = é atribuição
== é comparação

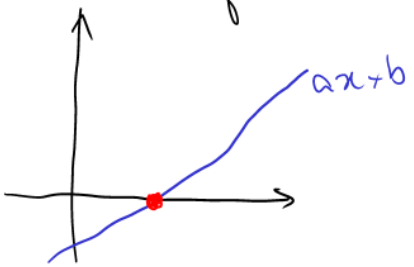


Dúvidas / comentários?

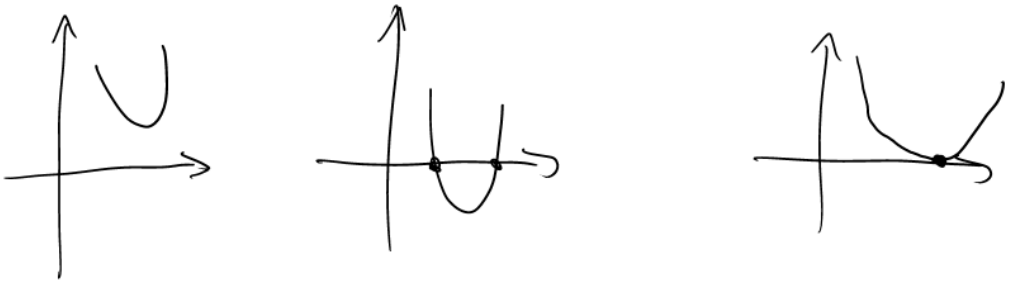
Exemplos:

1) Dados a, b, encontrar x tal que

$$ax + b = 0$$

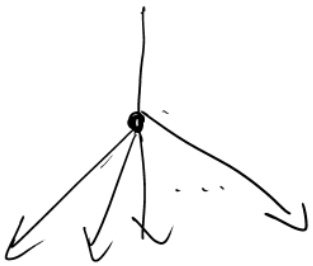


2) Dados a, b, c, determine quantas soluções reais tem a eq. $ax^2 + bx + c = 0$

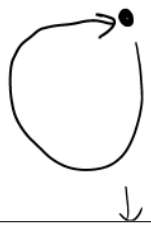


3) Dados a, b, c, encontrar as raízes (se existirem) de $ax^2 + bx + c = 0$

if / elif / elif / ... / else



while



Conjectura de Collatz: Começando em qualquer $x > 1$ inteiro e repetindo aplicações da função

$$C(n) = \begin{cases} n/2, & \text{se } n \text{ é par} \\ 3n+1, & \text{se } n \text{ é ímpar} \end{cases}$$

em algum momento chegamos a 1

↓

algo que achamos que é verdade mas não conseguimos provar ainda

---

Ex: Começando com $x = 4$

$C(4) = 4/2 = 2$

$C(2) = 2/2 = 1$ //

começando com $x = 5$

$C(5) = (3 \cdot 5) + 1 = 16$

$C(16) = 8$

$C(8) = 4$

$C(4) = 2$

$C(2) = 1$ //

Começando com $x = 3$

$C(3) = 10$

$C(10) = 5$

⋮

range: gerador de Progressão Aritmética

3 versões:

- range(n): gera $0, 1, 2, \ldots, n-1$
  - n: número de termos; começa em 0; razão 1
- range(inicial, quase_final): gera inicial, inicial+1, inicial+2, ..., quase_final−1
  - inicial: termo inicial, razão 1, (quase_final − inicial) termos

---

- range(inicial, quase_final, razão)
  - igual acima mas com a razão dada

Ex: Para gerar 7, 12, 17, 22, a chamada pode ser: range(7, 23, 5)

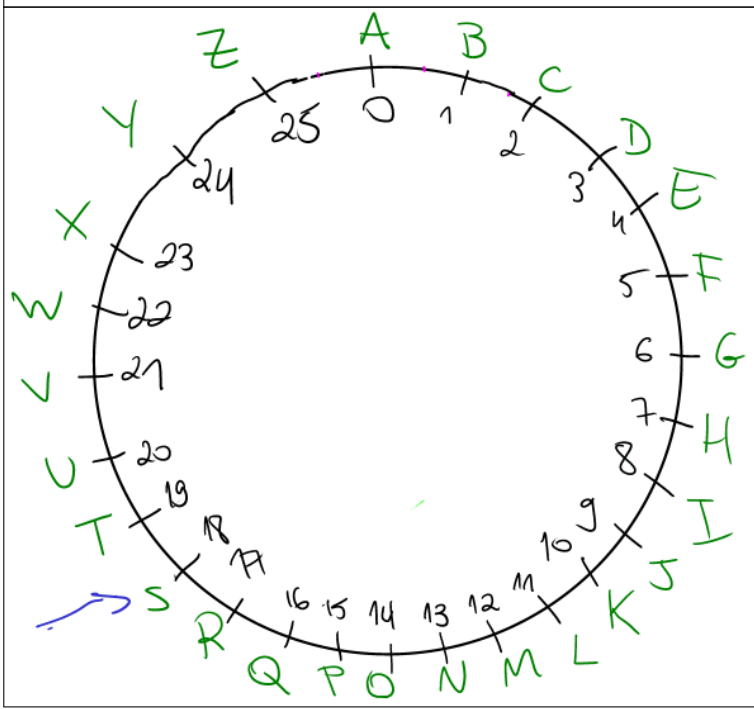
24
25
26
27

Ex: 5, −20, −45, −70, −95 : range(5, −20, −25)

Ex: 1, 2, 4, 8, 16 : não é range de ninguém

Dúvidas / comentários ?

Dúvidas & comentários



encriptar "CRIPTO"
usando chave = 10

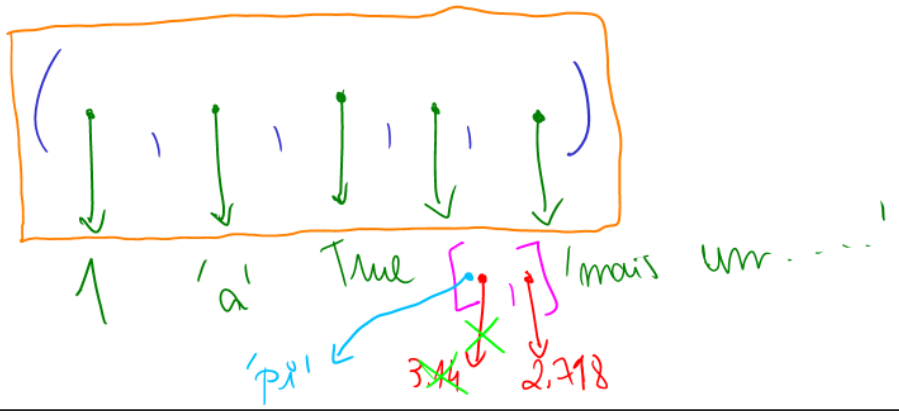
C R I P T O
↓ ↓
M

Dúvidas & comentários?

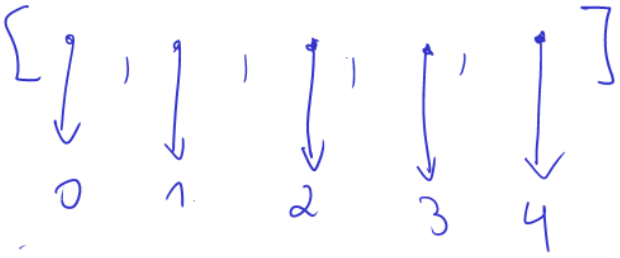
outra_tupla = (1, 'a', True, [3.14, 2.718], 'mais um elemento aqui')

Na memória

outra_tupla →

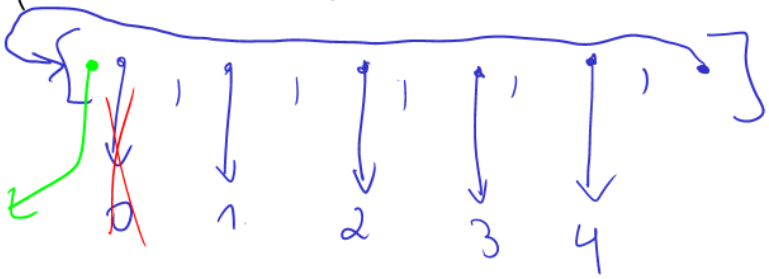


lista_doida →

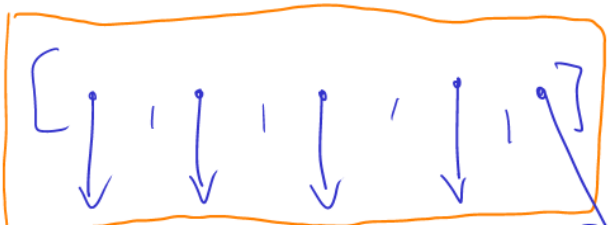


lista_doida.append(lista_doida)

lista_doida →



exemplo →



"OPA"

exemplo2 →

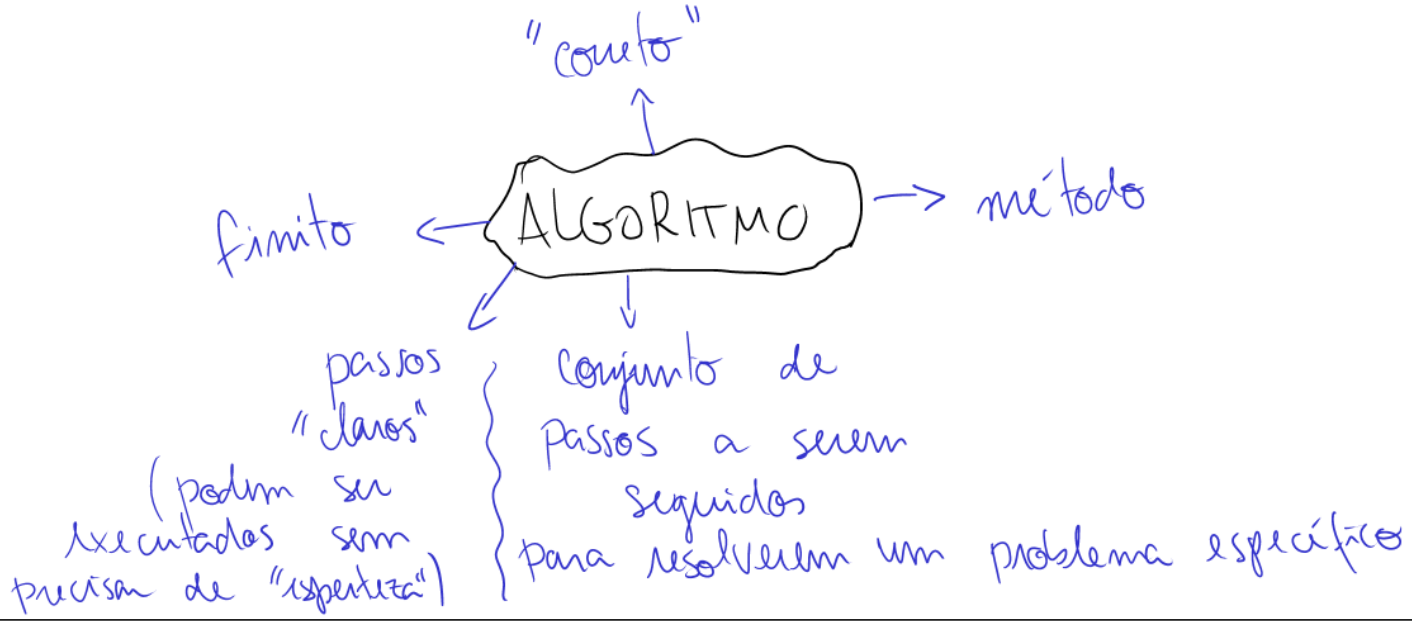
Al-Khwarizmi — al-jabr

$2x - 4 = 0$

$+4 \quad +4$

$2x = 4$

$x = 4/2 = 2$



Turing (1936), Church, Gödel/Herbrand

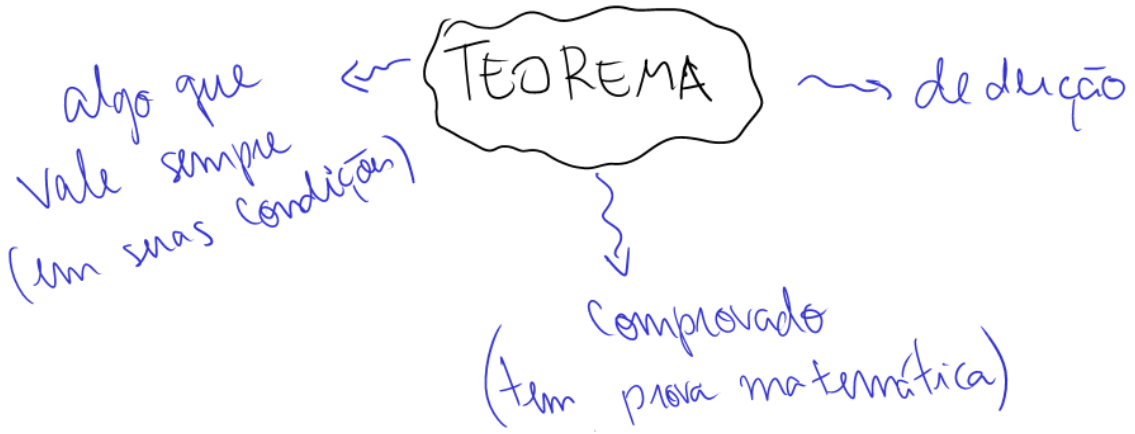
deram definições precisas, diferentes e equivalentes para "Algoritmo"

"Entscheidungsproblem": Problema da Decisão

Definição "de trabalho": um algoritmo é uma sequência finita de passos simples que, quando executada, resolve um problema dado.

A sequência pode precisar de dados extra para ser iniciada (entrada do alg.) e pode produzir dados ao seu final (saída do alg)

A descrição de um algoritmo (idealmente) deve ser tão precisa que o fato de que ele termina e resolve o problema possam ser provados (ou seja, são teoremas)



Definição "de trabalho": um teorema é uma afirmação (matemática) que tem uma prova aceita pela comunidade matemática.

prova ↔ um argumento

aceita → a prova deve convencer quem a lê

"Princípios" para uma boa prova:
- sem contradições
- sem ambiguidade → usando uma linguagem aceita e compreendida por todos
- "universalidade": cobre todos os casos do teorema

A linguagem da matemática:
- conectivos (e, ou, não, se ..., então ..., ... se, e somente se, ...)
- quantificadores (para todo, existe)

Além de teoremas e provas, as definições são muito importantes na matemática.

Definição: Seja n um número real.
Dizemos que n é par se existe um inteiro k tal que $n = 2 \cdot k$.
Um inteiro que não é par é chamado de ímpar.

Teorema: Seja n um ~~inteiro~~ real. → contexto (condição)
Se n é par, então n^2 é par. → afirmação (dentro do contexto)

Note: esse teorema não diz rigorosamente nada sobre números que não são inteiros, nem sobre inteiros n que sejam ímpares!

Prova: Seja n um ~~inteiro~~ real.
Suponha que n seja par. (restringindo o contexto)

Tarefa: Provar que n^2 é par, ou seja, provar que existe k inteiro tal que $n^2 = 2 \cdot k$. QUERO

Como n é par, existe l inteiro tal que
$n = 2 \cdot l$

Logo $n^2 = (2 \cdot l)^2$
$= 4 \cdot l^2$
$= 2 \cdot (2 \cdot l^2)$

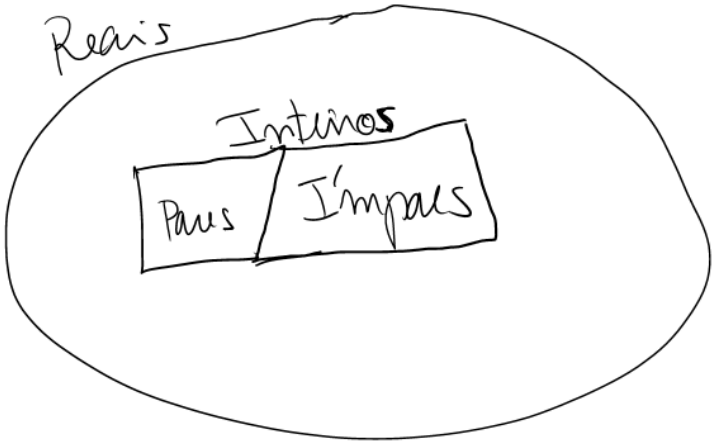
Como $2 \cdot l^2$ é um número inteiro (pois é o resultado da multiplicação de inteiros), concluímos que n^2 é par. □ ■ CQD QED

Dúvidas / comentários?

- Temos aulas 2ª (20/12) & 3ª (21/12)
  mas não 5ª (23/12)

Note:

Teorema: Seja n um número real.
Se n é par então n é inteiro.



A recíproca de um teorema cuja afirmação é "se A então B", em um certo contexto,

é a afirmação "se B então A" no mesmo contexto

(antes de ser provado, um teorema é chamado de conjectura). A recíproca do Teorema da última aula:

Conjectura 1: Seja n um real.
Se n^2 é par então n é par.

É falsa: vamos dar um contraexemplo (um

caso específico dentro do contexto onde a afirmação é falsa)

No nosso caso, para desprovar a Conjectura 1, podemos tomar o caso $n^2 = 2$, ou seja,

$n = \sqrt{2}$ ou $n = -\sqrt{2}$

(n é real e n^2 é par, mas n não é par)

Teorema
~~Conjectura~~ 2: Seja n um inteiro.
Se n^2 é par então n é par.

Rascunho n^2 par diz que existe k inteiro tal que $n^2 = 2 \cdot k$

Teorema: O produto de um inteiro por um par é par. (vamos aceitar como verdade)

Rascunho: $n^2 = n \cdot n$

Teorema: Todo inteiro é par ou ímpar (e nunca ambos!)

A contraposita de uma afirmação do tipo "se A então B" é a afirmação "se não B então não A". O valor de verdade é o mesmo!

Prova da Conjectura 2: Vamos provar a contrapositiva: "se n não é par então n^2 não é par"

Seja n inteiro e suponha que n não seja par.

Então n é ímpar, ou seja, existe inteiro l tal que $n = 2 \cdot l + 1$

Logo $n^2 = (2l+1)^2$

$= 4l^2 + 4l + 1$

$= 2(2l^2 + 2l) + 1$

Como $2l^2 + 2l$ é inteiro, n^2 é ímpar ▨

Rascunho: Quero mostrar que n^2 não é par, ou seja, n^2 é ímpar, ou seja, que existe p inteiro tal que $n^2 = 2p + 1$

—×—

Algoritmos

Exemplos: Como aprendemos a soma, subtrair, multiplicar, dividir no ensino básico.

Queremos falar de um caso importante: o Algoritmo de Euclides ou Euclidiano.

Def: Dados inteiros a & b, dizemos que

- a divide b
- a é divisor de b
- a é fator de b
- b é divisível por a
- b é múltiplo de a

} sinônimos

se existe inteiro c tal que $b = a \cdot c$

Em símbolos, escrevemos $a \mid b$,

Propriedades: (1) $\forall a \in \mathbb{Z} \ (a \mid 0)$

"para todo a inteiro"

(2) $\forall a \in \mathbb{Z} \ (a \neq 0 \rightarrow 0 \nmid a)$

"não divide"

(3) $\forall a \in \mathbb{Z} \ (1 \mid a)$

Prova: (1) Seja $a \in \mathbb{Z}$.

Quero provar: existe $c \in \mathbb{Z}$ tal que $0 = a \cdot c$

Como $0 = a \cdot 0$, temos $a \mid 0$ (no caso, $c = 0$)

(2) Seja $a \in \mathbb{Z}$ e suponha $a \neq 0$

Quero provar: $0 \nmid a$, ou seja, não existe $c \in \mathbb{Z}$ tal que $a = c \cdot 0$ ou $c \cdot 0 = a$

Para qualquer $c \in \mathbb{Z}$, temos $c \cdot 0 = 0 \neq a$

ou seja, não existe $c \in \mathbb{Z}$ tal que $c \cdot 0 = a$

Dúvidas / comentários?

"divide"

Propriedades de $|$: (4) $\forall a,b \in \mathbb{N} \left((b \neq 0 \wedge a|b) \rightarrow a \leq b\right)$

"e"

(cont.)

(5) $\forall a,b \in \mathbb{Z} \left((b \neq 0 \wedge a|b) \rightarrow |a| \leq |b|\right)$

(6) $\forall a,b \in \mathbb{Z} \left(a|b \text{ sse } |a| \mid |b|\right)$

"se e somente se"

Prova: (4): Sejam $a,b \in \mathbb{N}$. Suponha que $b \neq 0$ e $a|b$.

Como $a|b$, existe $c \in \mathbb{Z}$ tal que $b = a \cdot c$

Como $a,b \geq 0$, temos $c \geq 0$.

Como $b \neq 0$, temos $c \neq 0$.

"escondendo" provas por contradição

Logo $c \geq 1$.

Assim, como $a = \frac{b}{c}$ e $c \geq 1$, temos $a \leq b$.

Rascunho: Seria verdade para todos $a,b \in \mathbb{Z}$?

Falso $a = 1$, $b = -1$: $b \neq 0 \wedge a|b \wedge a \nleq b$

(6): Sejam $a,b \in \mathbb{Z}$

"ida" ($\Rightarrow$): Suponha $a|b$

Rascunho: Quero mostrar que $|a| \mid |b|$, ou seja, que existe $d \in \mathbb{Z}$ tal que $|b| = |a| \cdot d$

Como $a|b$, existe $c \in \mathbb{Z}$ tal que $b = a \cdot c$

Prova por casos

Caso 1: $a,b \geq 0$

Neste caso, temos $|a| = a$, $|b| = b$, portanto $|b| = |a| \cdot c$

Caso 2: $a < 0$, $b \geq 0$.

Neste caso temos $|a| = -a$, $|b| = b$

Assim, temos $(-a)\cdot(-c) = a\cdot c = b$

($|a|$ e $|b|$)

Caso 3: $a, b < 0$.

Neste caso temos $|a| = -a$, $|b| = -b$

$$\begin{aligned} |a|\cdot c &= (-a)\cdot c \\ &= -(a\cdot c) \\ &= -b \\ &= |b| \end{aligned}$$

Caso 4: $a \geq 0$, $b < 0$

Neste caso, temos $|a| = a$, $|b| = -b$

Portanto

$$\begin{aligned} |a|\cdot(-c) &= a\cdot(-c) \\ &= -(a\cdot c) \\ &= -b \\ &= |b| \end{aligned}$$

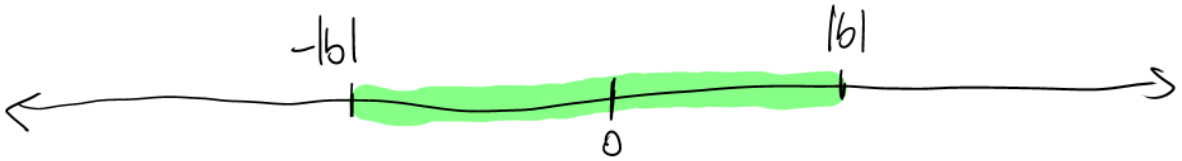
$\square(\Rightarrow)$

($\Leftarrow$) Exercício. ■

def: Dados inteiros a, b, definimos seu **mdc**, denotado $\text{mdc}(a,b)$, de acordo com os casos abaixo:

- Se $a = b = 0$, então definimos $\text{mdc}(a,b) = 0$
- Senão, $\text{mdc}(a,b)$ é o maior divisor comum de a & b (ou seja, o maior inteiro c com a propriedade que $c|a$ & $c|b$)

Note que, se $b \neq 0$, então seus divisores estão na "faixa"



O maior divisor de b (no caso $b \neq 0$) é $|b|$.

Portanto, se $a \neq 0$ ou $b \neq 0$, então

$$1 \leq mdc(a,b) \leq max(|a|,|b|)$$

Além disso, se $a \neq 0$ e $b \neq 0$, então

$$1 \leq mdc(a,b) \leq min(|a|,|b|)$$

Como calcular o $mdc(a,b)$?

Algoritmo "ingênuo" do mdc

Entrada: Inteiros a & b

Saída: $mdc(a,b)$

Passo 1: Se a == 0, retorne |b|.
Se b == 0, retorne |a|.
senão:

Passo 2: chute ← 1

Passo 3: Para i de 2 até $min(|a|,|b|)$:
se i|a & i|b:
chute ← i

Passo 4: Retorne chute

Exercícios:

provar que
- sempre termina
- é correto

Algoritmo de Euclides: "divisões sucessivas"!

Suponha $a, b > 0$.

Se $b \mid a$, então $mdc(a,b) = b$

$\uparrow \exists c \in \mathbb{N}(a = b \cdot c)$

Senão, se r_0 é o resto da div. de a por b, agora testamos se $r_0 \mid b$, pulo do gato!

Senão, se r_1 é o resto da div. de a por r_0, testamos se $r_1 \mid r_0$, e

$mdc(a, r_0) = r_0$ ← vamos montar

$\vdots$

$mdc(a,b)$

Assim sucessivamente.

$$a = b \cdot q_0 + r_0$$
$$b = r_0 \cdot q_1 + r_1$$
$$r_0 = r_1 \cdot q_2 + r_2$$
$$\vdots$$
$$r_{i-2} = r_{i-1} \cdot q_i + r_i$$

r_i → 0

r_{i-1} → $mdc(a,b)$

Exemplo: $a = 24$ $b = 18$

$24 = 18 \cdot 1 + 6$

$18 = 6 \cdot 3 + 0$

6 → $mdc(24,18)$

$a = 23$ $b = 34$

$23 = 34 \cdot 0 + 23$

$34 = 23 \cdot 1 + 11$

$23 = 11 \cdot 2 + 1$

$11 = 1 \cdot 11 + 0$

1 → $mdc(23,34)$

Dúvidas & comentários

- 2ª (20/12): aula de 13:00 às 14:00

Rascunho: para $a, b > 0$, se $b \mid a$ então $mdc(a,b) = b$

$\_\_ = \_ \cdot \_ + \_\_$

Algoritmo de Euclides

Entrada: $a, b \in \mathbb{Z}$

Saída: $mdc(a,b)$

Passo 1: Se $a = 0$, retorne $|b|$
Se $b = 0$, retorne $|a|$

Passo 2: dividendo $\leftarrow |a|$, divisor $\leftarrow |b|$
resto $\leftarrow$ dividendo % divisor

Passo 3: Enquanto resto $\neq 0$:

dividendo $\leftarrow$ divisor

divisor $\leftarrow$ resto

resto $\leftarrow$ dividendo % divisor

Passo 4: retorne divisor

Teorema (Terminação): Para quaisquer entradas $a, b \in \mathbb{Z}$, o algoritmo de Euclides termina após uma quantidade finita de passos.

Prova: Se $a = 0$ ou $b = 0$, o algoritmo para no primeiro passo.

Agora vamos assumir que $a \neq 0$ e $b \neq 0$.
Vamos chamar de $resto_0$, $resto_1$, $resto_2$, ... os valores da variável resto ao longo da execução.

Logo $resto_0 = |a| \,\%\, |b| \quad \in \mathbb{N}$

(se existir) $resto_1 = |b| \,\%\, resto_0 \quad \in \mathbb{N} \;\&\; resto_1 < resto_0$

(se existir) $resto_2 = resto_0 \,\%\, resto_1 \quad \in \mathbb{N} \;\&\; resto_2 < resto_1$

(se existir) $resto_3 = resto_1 \,\%\, resto_2 \quad \in \mathbb{N} \;\&\; resto_3 < resto_2$

⋮

Portanto, como $resto_0 > resto_1 > resto_2 > \cdots$, e todos os $resto_i$ são naturais (pois são restos de divisões de naturais por naturais), o processo tem que acabar em algum momento ■

Teorema (continuação)...

Note: para que seja verdade que o mdc entre os últimos valores de dividendo e divisor (esse é o valor retornado!) seja igual ao mdc entre os primeiros valores de dividendo e divisor, é <u>necessário</u> que o valor do mdc entre dividendo e divisor a qualquer momento do algoritmo seja o mesmo!

Então vamos provar isso.
Antes, um teorema "auxiliar"

Lema: Sejam $x, y, z, w \in \mathbb{Z}$.

Se $x = y \cdot z + w$

então $mdc(x, y) = mdc(y, w)$

Dúvidas & Comentários?

$n = 26$

$e = 7$

$77 \% 26 = 25$

$7 \cdot (11) = 77$

(11) não é o d que buscamos

77 cai na casa do 25

Euclides: exemplo mdc(120, 93)

dividendo	divisor	quociente	resto
120	93	1	27
93	27	3	12
27	12	2	3
12	3	4	0

mdc = 3

Para o algoritmo ter chance de estar certo, o valor de mdc(dividendo, divisor) tem que ser invariante (ou seja, não muda ao longo de toda a execução)

Na verdade, podemos provar algo ainda mais geral:

Lema: Sejam $x, y, z, w \in \mathbb{Z}$.

Se $x = y \cdot z + w$

então $mdc(x, y) = mdc(y, w)$

Dúvidas / comentários?

Prova da corretude do Alg. Euclides.

Vamos usar o seguinte lema (e depois vamos provar o lema):

Lema: Sejam $x, y, z, w \in \mathbb{Z}$.

Se $x = y \cdot z + w$

então $mdc(x, y) = mdc(y, w)$

Sejam $a, b \in \mathbb{Z}$ e vamos considerar a execução do Alg. Euclides com entradas a, b.

Se $a = 0$, o algoritmo retorna $|b|$, que é o valor de $mdc(a, b)$ [se $b = 0$, então $mdc(a, b) = 0 = |b|$, e se $b \neq 0$, então $|b|$ é o maior divisor de b, e também é divisor de $a = 0$, portanto $|b| = mdc(a, b)$]

Analogamente, se $b = 0$, então o valor retornado é $|a|$, que é $mdc(a, b)$.

Agora suponha que $a \neq 0$ & $b \neq 0$

Primeiramente, note que $mdc(a, b) = mdc(|a|, |b|)$

De fato • $mdc(a, b)$ é o maior número que divide a & b

• $mdc(|a|, |b|)$ é o maior número que divide $|a|$ & $|b|$

e (pela propriedade 6 da Aula 10), os números que dividem a & b são os mesmos que dividem $|a|$ & $|b|$.

Agora "estamos" nos Passos 2 & 3 do algoritmo.
Chamando de $dividendo_i$, $divisor_i$, $resto_i$
os valores de dividendo, divisor, resto
ao longo da execução do algoritmo, se a
execução fica no Passo 3 ("enquanto") k vezes,
então os últimos valores são

$$dividendo_k,\ divisor_k,\ resto_k$$

e o valor retornado pelo algoritmo é $divisor_k$.

(Nossa tarefa é argumentar que $mdc(a,b) = divisor_k$)

Se $k = 0$ (ou seja, não entramos no "enquanto" nenhuma vez), então $divisor_k = divisor_0 = |b|$.
Como $resto_0 = 0$ (pois não entramos no "enquanto"),
a divisão de $dividendo_0 = |a|$ por $divisor_0 = |b|$ é
exata, ou seja, $|b|$ divide $|a|$ e portanto
$mdc(|a|, |b|) = |b|$.
"
$mdc(a,b)$

Se $k > 0$, então as divisões feitas pelo algoritmo são

$$\begin{aligned}
dividendo_0 &= divisor_0 \cdot q_0 + resto_0 \\
dividendo_1 &= divisor_1 \cdot q_1 + resto_1 \\
dividendo_2 &= divisor_2 \cdot q_2 + resto_2 \\
&\vdots \\
dividendo_k &= divisor_k \cdot q_k + \cancel{resto_k} \to 0
\end{aligned}$$

Pelo Lema acima, para cada $i \leq k$, temos

$mdc(\text{dividendo}_i, \text{divisor}_i) = mdc(\text{divisor}_i, \text{resto}_i)$

Mas $\text{dividendo}_{i+1} = \text{divisor}_i$

$\text{divisor}_{i+1} = \text{resto}_i$

Logo $mdc(\text{dividendo}_i, \text{divisor}_i) = mdc(\text{dividendo}_{i+1}, \text{divisor}_{i+1})$

Ou seja, o valor do mdc entre "dividendo" e "divisor"

não muda ao longo do algoritmo!

Mas $mdc(\text{dividendo}_k, \text{divisor}_k) = \text{divisor}_k$

pois $\text{resto}_k = 0$, logo $mdc(a,b) = mdc(|a|,|b|)$

$= mdc(\text{dividendo}_0, \text{divisor}_0)$

$= \text{divisor}_k$ ∎

Falta provarmos o Lema!

Lema: Sejam $x, y, z, w \in \mathbb{Z}$.

Se $x = y \cdot z + w$ (★) ← isso é suficiente

então $mdc(x,y) = mdc(y,w)$ ← para isso

Prova: Se $x = 0 = y$, então $w = 0$, e então

$mdc(x,y) = 0 = mdc(y,w)$

Se $y = 0 = w$, então $x = 0$ e novamente

$mdc(x,y) = 0 = mdc(y,w)$

Então agora suponha que $\neg(x=0=y) \wedge \neg(y=0=w)$

Neste caso, por definição, temos

$mdc(x,y)$ é o maior inteiro que divide x & y

$mdc(y,w)$ é o maior inteiro que divide y & w.

(Tarefa: mostrar $mdc(x,y) = mdc(y,w)$)

Estratégia para tentar: $mdc(x,y)$ = o maior elemento de um conjunto A

$mdc(y,w)$ = o maior elemento de um conjunto B

Se provarmos $A=B$, então "de graça" teremos $mdc(x,y) = mdc(y,w)$

Provar que $A=B$, neste caso, seria provar que para qualquer inteiro d, temos $d \in A \iff d \in B$, ou seja:

para qualquer inteiro d: d divide x & y $\iff$ d divide y & w

---

Portanto, sejam d inteiro.

($\Rightarrow$) "ida" Suponha que $d|x$ & $d|y$.

(Tarefa: mostrar que ~~$d|y$~~ & $d|w$)

Lembrete: estamos sob a hipótese ($\star$): $x = y \cdot z + w$,

logo $w = x - y \cdot z$

Assim, se $x = d \cdot \alpha$ (pois $d|x$) & $y = d \cdot \beta$ (pois $d|y$) para algum par de inteiros α, β,

então $w = x - y \cdot z$

$= d \cdot \alpha - (d \cdot \beta) \cdot z$

$= d(\alpha - \beta z)$

ou seja, $d \mid w$. ($\alpha - \beta z$ é inteiro)

($\Leftarrow$) Análogo (exercício)
"volta"

■

Note: o Lema diz que a condição "$x = y \cdot z + w$" é suficiente para que $mdc(x,y) = mdc(y,w)$,

mas no momento não está claro se ela é necessária (voltaremos a isso em 2022)

— × —

Como vimos em aula anterior (sem provar), na cifra multiplicativa com alfabeto de tamanho n, uma chave de encriptação e "funciona", ou seja, pode ser descriptada, se e

Somente se $mdc(e,n) = 1$

No caso positivo, uma chave de descriptação é qualquer d que satisfaça

$$e \cdot d = n \cdot k + 1$$

para algum $k \in \mathbb{Z}$ (ou seja, $e \cdot d \,\%\, n = 1$)

Mas como encontrar d??

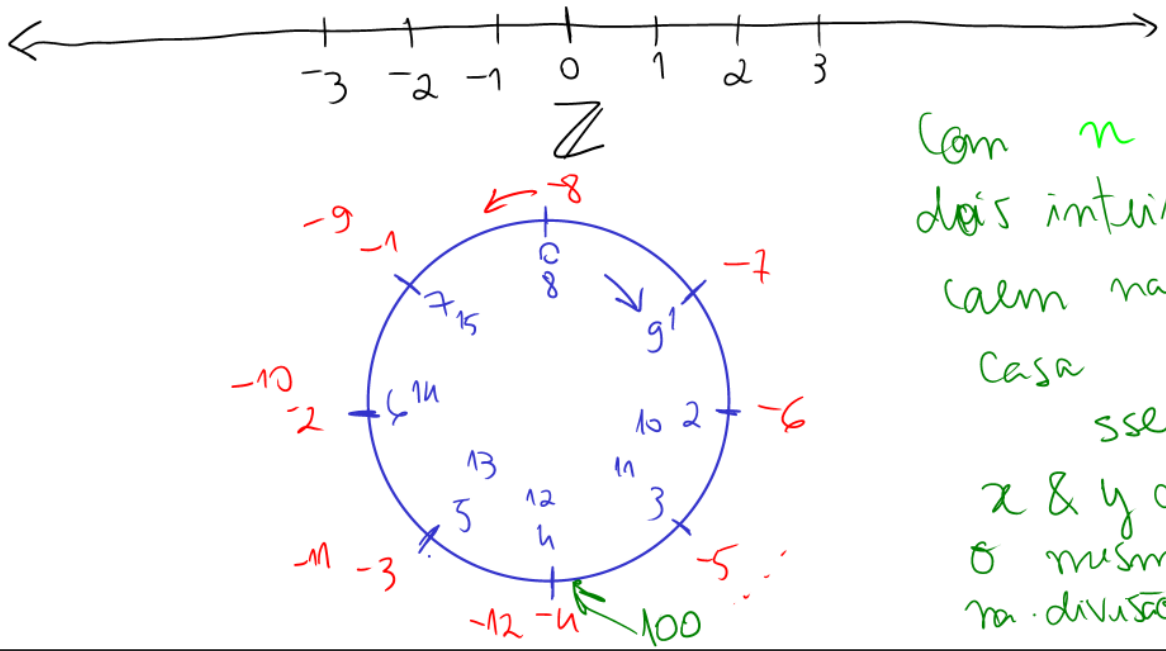
Na lista 1, exercício 4b, a ideia era força bruta.

Veremos uma forma mais eficiente, provando o seguinte teorema com um algoritmo eficiente

Teorema (Bézout): Para quaisquer inteiros a, b existem inteiros x, y tais que

$$a \cdot x + b \cdot y = mdc(a, b)$$

Vamos dar um algoritmo eficiente para encontrar



Com n posições, dois inteiros x, y caem na mesma casa sse x & y deixam o mesmo resto na divisão por n

Num mundo cíclico de tamanho $n > 0$, um número $e \in \mathbb{Z}$ tem inverso multiplicativo sse ⟵ "se e somente se"

$$mdc(e, n) = 1$$

No caso positivo, um inverso multiplicativo de e é qualquer $d \in \mathbb{Z}$ satisfazendo

$$\exists k \in \mathbb{Z} \left(e \cdot d = 1 + k \cdot n \right)$$

k: quantidade de voltas completas no ciclo

Ex: $n = 26$, $e = 5$, $d = 21$

pois

$$5 \cdot 21 = 1 + 4 \cdot 26$$

$n = 4$, $e = 5$, $d = 21$

pois

$$5 \cdot 21 = 1 + 26 \cdot 4$$

"Reassumindo" a frase anterior, fica: dados $e \in \mathbb{Z}$, $n \in \mathbb{Z}$, $n > 0$

$$\exists d, k \in \mathbb{Z} \left(e \cdot d + kn = 1 \right)$$

sse

$$mdc(e, n) = 1$$

Na verdade provaremos algo ainda mais geral:

Teorema (Bézout): Para quaisquer $a, b \in \mathbb{Z}$ existem $x, y \in \mathbb{Z}$ tais que $a \cdot x + b \cdot y = mdc(a, b)$

Provaremos o Teo. Bézout com um algoritmo que recebe como entrada a, b e retorna x, y.

↳ "prova construtiva"

Uma forma "jogão matemático" de enunciar o teorema poderia ser "$mdc(a,b)$ é uma combinação inteira de a & b"

A ideia é estender o Alg. Euclides (que calcula $mdc(a,b)$) para também calcular $x, y \in \mathbb{Z}$ tais que $ax + by = mdc(a,b)$

Relembrando Euclides (suponhamos $a, b > 0$)

$a = b \cdot q_0 + r_0$ $\quad x = 1$ $\quad y = -q_0$

$b = r_0 \cdot q_1 + r_1$

$r_0 = r_1 \cdot q_2 + r_2$

$r_1 = r_2 \cdot q_3 + r_3$

$\vdots$

$r_{i-3} = r_{i-2} \cdot q_{i-1} + r_{i-1}$ → quero escrever como comb. inteira de a&b

$r_{i-2} = r_{i-1} \cdot q_i + r_i$, $r_i = 0$

$\text{mdc}(a,b)$

Rascunho

$a = b \cdot q_0 + r_0 \rightsquigarrow r_0 = a + b \cdot (-q_0)$ ($x_0 = 1$, $y_0 = -q_0$)

$b = r_0 \cdot q_1 + r_1$

$r_1 = b - r_0 \cdot q_1$

$= b - (a + b \cdot (-q_0)) \cdot q_1$

$= a \cdot (-q_1) + b \cdot (1 + q_0 q_1)$ (x_1, y_1)

A próxima linha:

$r_0 = r_1 \cdot q_2 + r_2$

Quero x_2, y_2 tais que $a \cdot x_2 + b \cdot y_2 = r_2$

$r_2 = r_0 - r_1 \cdot q_2$

$= (a + b(-q_0)) - [a \cdot (-q_1) + b(1 + q_0 q_1)] \cdot q_2$

$= a[1 + q_1 q_2] + b[-q_0 - q_2 - q_0 q_1 q_2]$

(x_2, y_2)

Dúvidas & comentários

Rascunho

⋮ → no meio de uma execução de Euclides com entradas $a, b > 0$

$r_{k-2} = r_{k-1} \cdot q_k + r_k$

$r_{k-1} = r_k \cdot q_{k+1} + r_{k+1}$

$r_k = r_{k+1} \cdot q_{k+2} + r_{k+2}$

⋮

$r_k = a \cdot x_k + b \cdot y_k$

$r_{k+1} = a x_{k+1} + b \cdot y_{k+1}$

Calculando x_{k+2} & y_{k+2} tais que $r_{k+2} = a \cdot x_{k+2} + b \cdot y_{k+2}$:

$$r_{k+2} = r_k - r_{k+1} \cdot q_{k+2}$$

$$= (a x_k + b y_k) - (a x_{k+1} + b y_{k+1}) \cdot q_{k+2}$$

$$= a \underbrace{(x_k - x_{k+1} \cdot q_{k+2})}_{x_{k+2}} + b \underbrace{(y_k - y_{k+1} \cdot q_{k+2})}_{y_{k+2}}$$

Moral da história: se sei expressar dois restos consecutivos como combinações inteiras de a & b, o mesmo vale para os restos seguintes!

Pergunta: como começar?

Ideia: imaginar duas linhas "acima do começo"!

início → real

$r_{-2} = 1$ $\quad y_{-2} = 0$

$r_{-1} = 0$ $\quad y_{-1} = 1$

$a = b \cdot q_0 + r_0$ $\quad r_0$ $\quad y_0$

$b = r_0 \cdot q_1 + r_1$

$r_0 = r_1 \cdot q_2 + r_2$

Conferindo com a fórmula anterior:

$r_0 = r_{-2} - r_{-1} \cdot q_0 = 1$ $\quad \begin{cases} y_0 = y_{-2} - y_{-1} \cdot q_0 = -q_0 \end{cases}$

$r_1 = r_{-1} - r_0 \cdot q_1 = -q_1$ $\quad \begin{cases} y_1 = y_{-1} - y_0 \cdot q_1 = 1 + q_0 q_1 \end{cases}$

rascunho $\quad mdc(3,5) = 1$ $\quad 3 \cdot 2 + 5 \cdot (-1) = 1$ (x = 2, y = −1)

$mdc(-3,5) = 1$

Algoritmo Estendido de Euclides A.E.E. (versão preliminar)

Entradas: $a, b \in \mathbb{N}$

Saídas: $mdc(a,b)$, $x, y \in \mathbb{Z}$ tais que $ax + by = mdc(a,b)$

Passo 1: • Se $a = 0$, retorne $mdc(a,b) = b$, $x = 0$, $y = 1$

• Se $b = 0$, retorne $mdc(a,b) = a$, $x = 1$, $y = 0$

senão:

Passo 2: $x\_anterior \leftarrow 1$, $x\_atual \leftarrow 0$

$y\_anterior \leftarrow 0$, $y\_atual \leftarrow 1$

$Dividendo \leftarrow a$, $Divisor \leftarrow b$

Passo 3: Repita:

Quociente, Resto ← quociente e resto da divisão de Dividendo por Divisor

$x\_anterior, x\_atual \leftarrow x\_atual, x\_anterior - x\_atual \cdot Quociente$

$y\_anterior, y\_atual \leftarrow y\_atual, y\_anterior - y\_atual \cdot Quociente$

Se Resto = 0, retorne $mdc(a,b) = Divisor$

$x = x\_anterior$

$y = y\_anterior$

Senão: Dividendo, Divisor ← Divisor, Resto

Aplicação: Num alfabeto com 475 símbolos, é possível encriptar (multiplicando) usando chave... $e = 33$?

Se sim, qual chave descripta?

Resposta (Python): sim, $d = 72$

Pergunta: como tratar dos casos $a < 0$ ou $b < 0$?

Teorema (Terminação do AEE): ——————.

Prova: Porque o Alg. Euclides termina. (a condição de terminação é a mesma)

Teorema (Corretude do AEE): ——————.

Prova: O valor calculado como mdc está correto (Euclides). Além disso, pelas contas feitas anteriormente, a forma de atualizar os

Coeficientes a cada novo Resto calculado está correta ∎

Corolário: Teorema de Bézout!

↳ um teorema que é consequência direta de outro teorema já provado

Dúvidas & comentários?

Suponha $a, b \in \mathbb{N}$ e sejam $x, y \in \mathbb{Z}$ tais que $ax + by = mdc(a,b)$ ← podemos usar o AEE

Quais podem ser $z, w \in \mathbb{Z}$ tais que

$$(-a)\cdot z + bw = mdc(-a,b) \; ?$$

$z = -x$ $\Big\{ w = y$ ‖ ← Exercício $mdc(a,b)$

Algoritmo de Euclides Estendido ("versão completa")

Entradas: $a, b \in \mathbb{Z}$

Saídas: $mdc(a,b)$, $x, y \in \mathbb{Z}$ tais que $ax + by = mdc(a,b)$

Passo 1: $a' \leftarrow |a|$, $b' \leftarrow |b|$, $mult\_a = a'/a$, $mult\_b = b'/b$

Passo 2: $mdc(a',b')$, $x', y' \leftarrow$ saídas do AEE anterior com entradas a', b'

Passo 3: Retorne $mdc(a',b')$, $mult\_a * x'$, $mult\_b * y'$

Números Primos

def: Seja $p \in \mathbb{N}$. Dizemos que p é primo se ele tem exatamente dois divisores naturais. Um natural diferente de 0 & 1 que não seja primo é composto.

Os primos têm papel central na teoria dos números naturais, em parte porque são os "blocos fundamentais" de formação destes números:

Teorema (Fundamental da Aritmética, TFA, ou Teorema da Fatoração Única):

Seja $n \in \mathbb{N}$, $n \geq 2$.

1) Então n pode ser escrito como produto de primos de uma única forma "a menos de ordenação"

2) Então n pode ser escrito como produto de primos em ordem não-decrescente de uma única forma

3) Então existem únicos

- $k \in \mathbb{N}$
- $p_0 < p_1 < p_2 < \dots < p_{k-1}$ primos
- $e_0, e_1, e_2, \dots, e_{k-1} > 0$ naturais

tais que $n = p_0^{e_0} \cdot p_1^{e_1} \cdot p_2^{e_2} \cdot \dots \cdot p_{k-1}^{e_{k-1}}$

produtório $\rightarrow = \prod_{i=0}^{k-1} p_i^{e_i}$

Ex: $12 = 2 \cdot 3 \cdot 2 = 3 \cdot 2 \cdot 2 = 2 \cdot 2 \cdot 3 = 2^2 \cdot 3 = 2^2 \cdot 3 \cdot 2^3$

$27 = 3^3$

$= 2^0 \cdot 3^3$

$= 3^3 \cdot 5^0$

$= 2^0 \cdot 3^3 \cdot 5^0 \cdot 101^0$

Vamos provar o TFA em sua forma (3).

Como padrão, separaremos a prova em duas partes: existência e unicidade.

↳ faremos um algoritmo

Rascunho: "$\exists! x\,(\varphi(x))$" é o mesmo que

"$\exists x\,[\varphi(x) \wedge \forall y\,(\varphi(y) \to y = x)]$"

≥ 1 existência — unicidade ≤ 1

ou

$$\exists x\,[\varphi(x) \wedge \neg\exists y\,(y \neq x \wedge \varphi(y))]$$

$$\exists x\,[\varphi(x) \wedge \forall y\,\neg(y \neq x \wedge \varphi(y))]$$

$$\exists x\,[\varphi(x) \wedge \forall y\,(\varphi(y) \to y = x)]$$

Existência

Algoritmo: dado $n \geq 2$ natural, a partir de $d = 2$, teste se d divide n. Se sim, "guarde" d e troque n por n/d se não, troque d por $d+1$. Pare quando $n = 1$.

$150 = 2 \cdot 75$

$= 2 \cdot 3 \cdot 25$

$= 2 \cdot 3 \cdot 5 \cdot 5$

$= 2 \cdot 3 \cdot 5^2 \cdot 1$ fim

Exemplo: $n = 300$

$d = 2$ divide! $n = 150$

$d = 2$ divide! $n = 75$

$d = 2$ não divide

$d = 3$ divide! $n = 25$

$d = 3$ não divide

$d = 4$ não divide

$d = 5$ divide! $n = 5$

$d = 5$ divide! $n = 1$ FIM

$\{2:2,\ 3:1,\ 5:2\}$

Comentários pré-conclusão:

Como estamos "guardando" qualquer d que divida n ao longo do algoritmo, precisaremos mostrar que apenas d's primos dividem n ao longo do algoritmo

Dúvidas & comentários?

É verdade que

✓ "existe no máximo 1 primo par"

&

✓ "existe pelo menos 1 primo par"

(unicidade ↖ existência ↙)

✓ "existe exatamente 1 primo par"

Algoritmo de Fatoração em Primos

Entrada: $n \in \mathbb{N}$, $n \geq 2$

Saída: primos $p_0 < p_1 < \cdots < p_{k-1}$
expoentes $e_0, e_1, \ldots, e_{k-1} > 0$
tais que $n = \prod_{i=0}^{k-1} p_i^{e_i}$

Passo 1: $F \leftarrow n$

Passo 2. Para d de 2 até n (inclusive):
Enquanto $d \mid F$:

$F \leftarrow F/d$

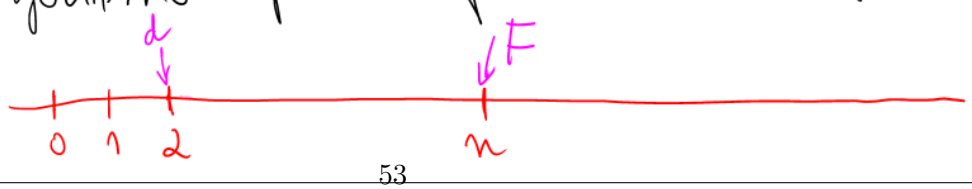
"guarde d ou aumente seu expoente na fatoração"

Se F for 1, retorne a fatoração encontrada.

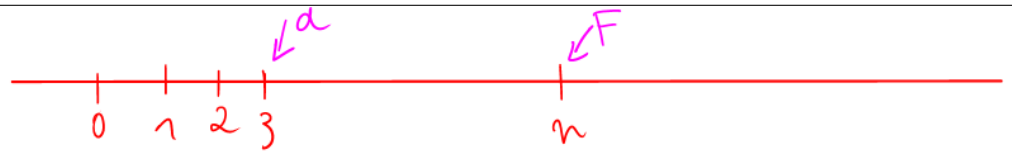
Teorema (Terminação): _____.

Prova: O algoritmo para quando $F = 1$.

Rascunho:
no início

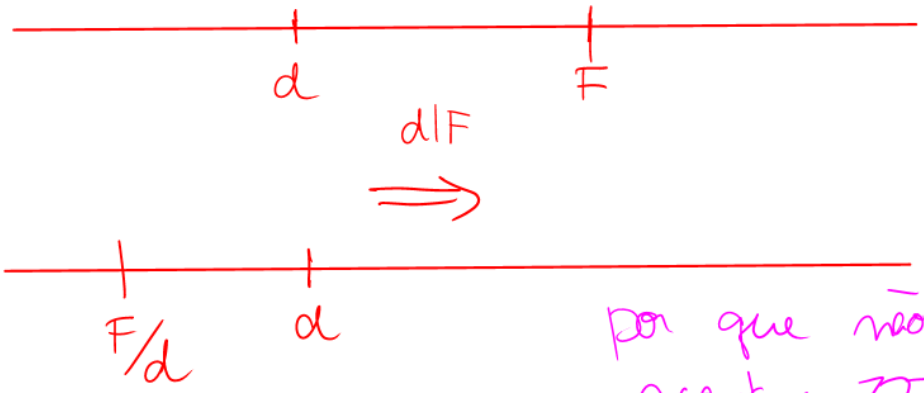


quando $d \nmid F$:



quando $d \mid F$:

Temos $F=1$ quando (imediatamente antes) tivemos $d=F$, e portanto atualizamos $F \leftarrow F/d$, ou seja $F \leftarrow 1$. (prova continua mais abaixo).



por que não acontece???

$d=7 \quad F=35$

$F/d=5<7$

Lema 1: A qualquer momento de uma execução do algoritmo, o valor de F é divisor dos valores anteriores de F

Prova: O valor de F, quando atualizado, é trocado pelo resultado da divisão do valor anterior por um divisor dele, i.e.,

$$F_{novo} = F_{antigo}/d,$$

logo $F_{novo} \mid F_{antigo}$ ■(Lema)

Corolário: Quando atualizamos d, o valor antigo não será divisor de nenhum valor futuro de F.

Prova: Atualizamos d quando $d \nmid F$, e valores futuros de F dividem o valor atual de F. (pelo Lema 1) ■ (corolário)

Lema 2: A qualquer momento de uma execução do algoritmo, d é menor ou igual ao menor fator de F (maior do que 1).

Prova: Suponha, para chegarmos a uma contradição, que em algum momento tenhamos que d é maior que o menor fator de F maior que 1 (digamos que tal fator seja m).

$$\left(2 \leq m < d \leq F \quad \text{com} \quad m \mid F.\right)$$

Como $2 \leq m$ e $m < d$, o valor de d foi m em algum momento do passado.

Como d não vale m no momento atual, ele foi "atualizado" de m para $m+1$ no passado, e pelo corolário acima temos que m não divide nenhum valor de F a partir de então, o que contradiz a hipótese de que m divide o valor atual de F. ↯ ← contradição desejada

Portanto $d \leq m$. ■ (Lema 2)

Corolário: Ao atualizarmos F, temos duas opções:

- o novo valor é $\geq d$
- o novo valor é 1

Prova (Terminação, cont.):

Quando atualizamos d, mantemos $d \leq F$ mas a distância $F-d$ diminui.

Quando atualizamos F, ou fazemos $F=1$ (e fim!) ou mantemos $d \leq F$ e diminuímos a distância $F-d$.

Portanto em algum momento teremos $F=d$, e no próximo passo F será 1 (e fim!) ■

Teorema (Corretude): ——————————

Prova: Por construção, a saída é uma fatoração do n dado na entrada, pois guardamos d apenas quando este divide F (e F sempre divide n, pelo Lema 1 acima), e então atualizamos $F_{novo} = F_{antigo}/d$

Logo, se $F_0, F_1, F_2, \ldots$ são os valores de F ao longo da execução e $d_0, d_1, d_2, \ldots$ os valores de d correspondentes temos

$$n = F_0$$

$$F_1 = F_0/d_0, \quad F_2 = F_1/d_1 = n/d_0 \cdot d_1$$

$$F_3 = F_2/d_2 = \frac{n}{d_0 \cdot d_1 \cdot d_2}$$

$$1 = F_i = \frac{n}{d_0 \cdot d_1 \cdot \ldots \cdot d_{i-1}}$$

Falta provarmos que os fatores "guardados" são primos, ou seja, se ao longo da execução temos $d \mid F$ então d é primo, ou seja (contrapos.) se d é composto então $d \nmid F$.

Já sabemos (lema 2) que d é menor ou igual ao menor fator de F (maior que 1), logo se $d \mid F$ então d é o menor fator de F (maior que 1)

Lema 3: Para qualquer natural $k \geq 2$, o menor fator natural de k (exceto 1) é primo.

Prova: Queremos provar que se $m > 1$ é o menor natural > 1 que divide k, então m é primo.

A contrapositiva: se $m > 1$ é composto então m não é o menor natural > 1 que divide k.

Provando.

Se $m > 1$ é composto, então $m = x \cdot y$ com $x, y \in \mathbb{N}$ e $1 < x, y < m$

fator cofator

Assim, se $m \mid k$, como $x \mid m$ & $y \mid m$, temos que ambos x & y são fatores de k, $x, y > 1$ e $x, y < m$.

Portanto m não é o menor fator >1 de R. ▨ (Lema 3)

Prova (corretude, cont.): "Guardamos" d quando ele é o menor fator >1 do F daquele momento. Pelo Lema 3, neste caso d é primo. ▨ (corretude)

▨ (Existência TFA)

Dúvidas & comentários?

L2 Q1 c) $\forall a,b,c \in \mathbb{Z} \left((a|b \wedge b|c) \rightarrow a|c \right)$

ou seja quero provar: $\exists z \in \mathbb{Z}$ tal que $c = a \cdot z$

Prova: Sejam $a, b, c \in \mathbb{Z}$

Suponha $a|b \wedge b|c$. Quero provar: $a|c$

Como $a|b$, temos que $\exists x \in \mathbb{Z}$ tal que $b = a \cdot x$ (1)

Como $b|c$, temos que $\exists y \in \mathbb{Z}$ tal que $c = b \cdot y$ (2)

Substituindo (1) em (2), temos $c = (a \cdot x) \cdot y = a \cdot x \cdot y$

Logo, como $x \cdot y$ é inteiro, temos $a|c$. ∎

Rascunho

$16/2 = 8$

$16 = 8 \cdot 2$

$19 = 1 \cdot 19$

$19/1 = 19$ & $19/19 = 1$

$\forall x \in \mathbb{Z} \left(x/1 = x \wedge x/x = 1 \right)$ ← primos só têm esses divisores naturais.

Vamos otimizar um pouco o alg. de fatoração em primos.

Rascunho: suponha que o menor fator $m > 1$ de F satisfaça $m > 5$

Como $m|F$, existe $x \in \mathbb{N}$ tal que

$F = m \cdot x$ → x é o maior fator de F (exceto o próprio F)

↘ cofator

Logo o maior fator $x < F$ de F satisfaz

$x < F/5$

$F = m \cdot x \Rightarrow 5 < m = F/x \Rightarrow x < F/5$

Ou seja: não há fator de F entre $F/5$ & F

No geral: se o menor fator $m > 1$ de F satisfaz $m > k$ para algum $k \in \mathbb{N}$, então o maior fator $x < F$ de F satisfaz

$$x < F/k,$$

e portanto entre F/k e F não há fatores de F.

Proposta: parar de procurar quando $\lfloor F/d \rfloor \leq d$ ou

"piso" (parte inteira)

$d^2 \geq F$ ou $d \geq \sqrt{F}$

↑ equivalente matematicamente mas para progamação pode haver erro de ponto flutuante na $\sqrt{F}$

$\frac{F}{d} \leq \frac{d}{1}$

$F \leq d \cdot d$

Lema: Para qualquer $k \in \mathbb{N}$, $k \geq 2$, se k é composto então seu menor fator $m > 1$ satisfaz

$$m \leq \lfloor \sqrt{k} \rfloor$$

Prova: Suponha k composto. Logo há fatores de k diferentes de 1 & k. Como m é inteiro, basta provar

$$m \leq \sqrt{k}.$$

Como m/k, existe $x \in \mathbb{Z}$ tal que $k = m \cdot x$ → é o maior fator $< k$

Mas então, como $1 < m < k$, temos $1 < x < k$ e x é um fator de k.

Como m é o menor fator de k (exceto 1), temos

$$x \geq m$$

Logo $$k = m \cdot x \Rightarrow m = \frac{k}{x} \leq \frac{k}{m}$$

Portanto $m^2 \leq k$, ou seja, $m \leq \sqrt{k}$. ▨

Exercício para praticar: escrever o algoritmo "otimizado" de fatoração em primos e provar sua terminação & corretude.

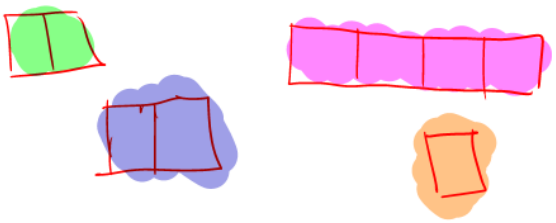
Unicidade (TFA)

Rascunho: estratégia para provar que "existe no máximo um x satisfazendo $\varphi(x)$":

- provar que $\forall x, y \; ((\varphi(x) \wedge \varphi(y)) \rightarrow x = y)$ ⟸
- provar que $\neg \exists x, y \; (x \neq y \wedge \varphi(x) \wedge \varphi(y))$

Note que a "unicidade" tem que vir de alguma propriedade especial dos primos; se não exigíssemos primos na fatoração, não teríamos unicidade. Por exemplo

$$100 = 2 \cdot 50 = 4 \cdot 25 = 5 \cdot 20$$



Provaremos:

Teorema (Propriedade Fundamental dos Primos):
Se p é primo e $x, y \in \mathbb{Z}$ tais que $p \mid (x \cdot y)$, então $p \mid x$ ou $p \mid y$

Dúvidas & comentários?

"tem a PFP como corolário"

Provaremos algo mais forte do que a P.F.P.:

Lema (§2.6): Sejam $a, b, c \in \mathbb{Z}$ com $\text{mdc}(a,b) = 1$. (coprimos ou primos entre si)

1) Se $b \mid (a \cdot c)$ então $b \mid c$

2) Se $a \mid c$ & $b \mid c$ então $(a \cdot b) \mid c$

Primeiro vejamos como com o Lema provamos a PFP.

Prova (P.F.P. | assumindo o Lema).

Suponha p primo, $x, y \in \mathbb{Z}$ tais que $p \mid (x \cdot y)$.

Tarefa: Provar $p \mid x$ ou $p \mid y$, ou equivalentemente

$$p \nmid x \longrightarrow p \mid y$$

Suponha $p \nmid x$

Tarefa: $p \mid y$

Para usarmos o item (1) do Lema, precisamos

---

mostrar que $\text{mdc}(x, p) = 1$

Como $\text{mdc}(x,p) \mid x$ e $p \nmid x$, temos $\text{mdc}(x,p) \neq p$.

Como $\text{mdc}(x,p) \mid p$ e p é primo, temos $\text{mdc}(x,p) \in \{1, p\}$

Logo, juntando os dois fatos, $\text{mdc}(x,p) = 1$.

Agora podemos usar o Lema: como $p \mid x \cdot y$ e $\text{mdc}(x,p) = 1$, concluímos $p \mid y$. ▨

Agora provemos o Lema.

Prova (Lema). Sejam $a, b, c \in \mathbb{Z}$ com $mdc(a,b) = 1$.

1) Suponha que $b \mid ac$.

Tarefa: Provar que $b \mid c$

Pelo Teorema de Bézout, existem $x, y \in \mathbb{Z}$ tais que

$$a \cdot x + b \cdot y = 1 \rightarrow mdc(a,b)$$

Multiplicando por c, fica

$$acx + bcy = c$$

Como $b \mid ac$, temos que $b \mid (acx + bcy)$

ou seja, $b \mid c$.

2) Suponha $a \mid c$ & $b \mid c$

Tarefa: Provar $ab \mid c$

Como $a \mid c$, temos $c = a \cdot q$ para algum $q \in \mathbb{Z}$

$\rightsquigarrow c = b \cdot q'$ para algum $q' \in \mathbb{Z}$

Note: $b \mid c$, logo $b \mid a \cdot q$

Como $mdc(a,b) = 1$, pelo item (1) concluímos $b \mid q$.

Logo existe $r \in \mathbb{Z}$ tal que $q = b \cdot r$.

Portanto $c = a \cdot q$

$= a \cdot b \cdot r$

Logo $ab \mid c$ ▨

Podemos finalmente provar a "unicidade" no TFA.

Prova (Unicidade no TFA): Seja $n \in \mathbb{N}$, $n \geq 2$.

Sejam:

- $k \in \mathbb{N}$
- $p_0 < p_1 < \dots < p_{k-1}$ primos
- $e_0, e_1, \dots, e_{k-1} > 0$ naturais

tais que $n = \prod_{i=0}^{k-1} p_i^{e_i}$

&

- $l \in \mathbb{N}$
- $q_0 < q_1 < \dots < q_{l-1}$ primos
- $f_0, f_1, \dots, f_{l-1} > 0$ naturais

tais que $n = \prod_{j=0}^{l-1} q_j^{f_j}$

Tarefa: Provar que $k = l$, $p_0 = q_0$, $p_1 = q_1, \dots$
$e_0 = f_0$, $e_1 = f_1, \dots$

Temos $\prod_{i=0}^{k-1} p_i^{e_i} = n = \prod_{j=0}^{l-1} q_j^{f_j}$

Considere p_i com $0 \leq i < k$.

Como $e_i > 0$, temos $p_i \mid n$, ou seja, $p_i \mid \prod_{j=0}^{l-1} q_j^{f_j}$

Logo, pela PFP, existe $j \in \mathbb{N}$ com $0 \leq j < l$, tal que $p_i \mid q_j$

Como p_i & q_j são primos, concluímos $p_i = q_j$

Logo: todos os primos da fatoração da esquerda aparecem na fatoração da direita.

Portanto $k \leq l$

Analogamente, podemos concluir $l \leq k$ (Exercício)

Assim $\underline{k = l}$

Além disso, como $p_0 < p_1 < \dots < p_{k-1}$
& $q_0 < q_1 < \dots < q_{k-1}$

necessariamente devemos ter $\underline{p_0 = q_0}$, pois $p_0 = q_j$ para algum j, mas se $j > 0$, então não haveria i tal que $p_i = q_0$.

Analogamente temos $\underline{p_1 = q_1, p_2 = q_2}$, etc.

Falta mostrar: $e_0 = f_0$, $e_1 = f_1$, ...

Consideremos e_i com $0 \leq i < k$

Temos $n / p_i^{e_i} \in \mathbb{N}$

Mas $$n = \prod_{j=0}^{l-1} q_j^{f_j} = \prod_{j=0}^{k-1} p_j^{f_j}$$

$$\frac{2^5 \cdot 3 \cdot 5^4 \cdot 7^{10}}{5^3} = 5^{4-3} \cdot 2^5 \cdot 3 \cdot 7^{10}$$

Portanto $$\frac{n}{p_i^{e_i}} = \frac{\prod_{j=0}^{k-1} p_j^{f_j}}{p_i^{e_i}} = p_i^{(f_i - e_i)} \cdot \prod_{\substack{j=0 \\ j \neq i}}^{k-1} p_j^{f_j}$$

← excluindo o índice i

Como $\frac{n}{p_i^{e_i}} \in \mathbb{N}$, temos $f_i - e_i \geq 0$, ou seja $f_i \geq e_i$.

Analogamente (exercício) temos $e_i \geq f_i$, logo $e_i = f_i$ ∎

Rascunho: $p_i^{-e_i} \cdot \prod_{j=0}^{k-1} p_j^{f_j} = \left(p_i^{-e_i} \cdot p_i^{f_i}\right) \cdot \prod_{\substack{j=0 \\ j \neq i}}^{k-1} p_j^{f_j}$

$p_i^{(f_i - e_i)}$

Exemplo: Digamos que $a, b \geq 2$ naturais, com

$$a = \prod_{i=0}^{k-1} p_i^{e_i} \qquad b = \prod_{j=0}^{l-1} q_j^{f_j}$$

1) Olhando para as fatorações, como decidir se $a \mid b$?

2) Como são as fatorações de $\text{mdc}(a,b)$ & $\text{mmc}(a,b)$?

Dúvidas & comentários

Bézout: $\forall a,b \in \mathbb{Z}\ \exists\, x,y \in \mathbb{Z}\ \left(ax + by = mdc(a,b)\right)$

prova disso: AEE

$\neg\left(\forall x,y,z \in \mathbb{Z}\left(x \mid yz \quad \text{sse} \quad (x \mid y \text{ ou } x \mid z)\right)\right)$

$\exists x,y,z \in \mathbb{Z}\left(\neg\left(x \mid yz \quad \text{sse} \quad (x \mid y \text{ ou } x \mid z)\right)\right)$

"é equivalente a"

$\neg(A \to B)$: A é V & B é F

$ax + by = mdc(a,b)$

"tem uma propriedade P"

também vale P

Como já comentamos, a segurança do RSA e outros métodos similares de criptografia depende da dificuldade do problema de

fatoração, o que está relacionado à existência de números primos "grandes". Neste sentido, o seguinte resultado traz um pouco de paz:

Teorema (Euclides): Existem infinitos números primos

Propostas de formalização:

- $\forall p$ primo $\exists q$ primo $(q > p)$ — Nicolas & Isac
- $\forall n \in \mathbb{N}$ $\exists$ p primo $(p > n)$
- $\forall n \in \mathbb{N}$ ("existem pelo menos n primos distintos")
- $\forall n \in \mathbb{N}$ $\neg$ ("a quantidade de primos é exatamente n"))

"p é primo": • $\forall n \in \mathbb{N} \left((n \neq 1 \wedge n \neq p) \rightarrow n \nmid p \right)$ — Mariana

• $\forall q \in \mathbb{N} \left(q < p \rightarrow mdc(p, q) = 1 \right)$ — Pedro Arthur

Prova: Vamos provar o teorema em sua versão "$\forall p$ primo $\exists q$ primo $(q > p)$".

Seja p primo.

Sejam $p_0 < p_1 < p_2 < \cdots < p_n$ os primos até p.
($p_0 = 2$, $p_1 = 3$, $p_n = p$)

Seja $x = \left(\prod_{i=0}^{n} p_i \right) + 1$. (*) Note que $x \in \mathbb{N}$

Afirmação: $x \geq 3$

De fato, $\prod_{i=0}^{n} p_i \geq p \geq 2$, logo $x \geq 3$.

Então, pelo Lema 3 da Aula 17, o menor fator q de x (exceto 1) é um número primo.

Afirmação: Nenhum p_i, com $0 \leq i \leq n$, divide x.

De fato, a divisão de x por p_i, com $0 \leq i \leq n$, tem quociente $\left(\prod_{j<i} p_j \cdot \prod_{\substack{j>i \\ j \leq n}} p_j\right)$ e resto 1, por $(\star)$

Logo, como q é primo e $q \mid x$, concluímos que $q \neq p_i$ para todo $i \leq n$.

rascunho $p = 5$

$x = 2 \cdot 3 \cdot 5 + 1$

$= 31$

Como os p_i's são todos os primos $\leq p$, concluímos que $q > p$. ∎

Pergunta "motivadora": Como definir a sequência de números primos?

$p_0 = 2 \quad p_1 = 3 \quad p_2 = 5, \quad p_3 = 7, \quad p_4 = 11, \ldots,$

$p_n =$ "o menor primo $> p_{n-1}$"

Se fosse a sequência das potências de 2

$x_0 = 1$, $x_1 = 2$, $x_2 = 4$, $x_3 = 8$, ...

$x_n = 2^n$

Se fosse a sequência de Fibonacci?

$F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$,

$F_5 = 5$, $F_6 = 8$, $F_7 = 13$, ..., $F_n = F_{n-2} + F_{n-1}$

Problema: $y_0 = 22$, $y_1 = 6$, $y_2 = 17$, ...

$y_n = 3 + y_{n+1}$

Dúvidas & comentários ?

A função de Fibonacci $F: \mathbb{N} \to \mathbb{N}$

$$F(n) = \begin{cases} 0, & \text{se } n = 0 \\ 1, & \text{se } n = 1 \\ F(n-2) + F(n-1), & \text{se } n \geq 2 \end{cases}$$

casos base

Consigo calcular $F(8)$? "de baixo para cima"

$F(2) = F(0) + F(1) = 0 + 1 = 1$ $\quad F(3) = F(1) + F(2) = 1 + 1 = 2$

$F(4) = F(2) + F(3) = 1 + 2 = 3$ $\quad F(5) = F(3) + F(4) = 2 + 3 = 5$

⋮

$F(8) = \ldots$

"de cima para baixo"

$F(8) = F(6) + F(7)$

$= (F(4) + F(5)) + (F(5) + F(6))$

$= [F(2) + F(3)] + [F(3) + F(4)] + [F(3) + F(4)] + [F(4) + F(5)]$

$= \ldots$

= uma expressão contendo apenas $F(0)$ & $F(1)$

A "função dos números primos" $P: \mathbb{N} \to \mathbb{N}$

$$P(n) = \begin{cases} 2, & \text{se } n = 0 \\ \text{"o menor número primo", que é maior do que } P(n-1), & \text{se } n \geq 1 \end{cases}$$

caso base

Consigo calcular $P(5)$?

↑:

$P(1) = 3$, $P(2) = 5$, $P(3) = 7$

$P(4) = 11$, $P(5) = 13$

↓: $P(5)$ = o menor primo maior que $P(4)$

= o menor primo maior que (o menor primo maior que $P(3)$)

= o menor primo maior que (o menor primo maior que (o menor primo maior que $P(2)$))

= ...

= expressão contendo apenas o caso base $P(6)$

O exemplo y da aula 20

$$y : \mathbb{N} \to \mathbb{N}$$

$$y(n) = \begin{cases} 22, & \text{se } n = 0 \\ 6, & \text{se } n = 1 \\ 17, & \text{se } n = 2 \\ 3 + y(n+1), & \text{se } n \geq 3 \end{cases}$$

Posso calcular $y(6)$?

↑: $y(3) = 3 + y(4) = 3 + (3 + y(5)) = 3 + (3 + (3 + y(6))) = \ldots$

↓: $y(6) = 3 + y(7) = 3 + (3 + y(8)) = 3 + (3 + (3 + y(9))) = \ldots$

Exemplo esquisito $H : \mathbb{N} \to \mathbb{N}$

$$H(n) = \begin{cases} 20, & \text{se } n = 7 \\ 43, & \text{se } n \neq 7 \text{ e } n \text{ é ímpar} \\ 15 + 2 \cdot H(2n+1), & \text{se } n \text{ é par} \end{cases}$$

Casos base

Como calcular $H(10)$?

↓: $H(10) = 15 + 2H(21)$

$= 15 + 2 \cdot 43$

$= 101$

Exemplo: a "função de Collatz" $C: \mathbb{N} \to \mathbb{N}$

$$C(n) = \begin{cases} 1, & \text{se } n \leq 1 \quad \} \text{ caso base} \\ C(n/2), & \text{se } n \geq 2 \text{ e } n \text{ é par} \\ C(3n+1), & \text{se } n > 2 \text{ e } n \text{ é ímpar} \end{cases}$$

Posso calcular $C(10)$???

↓: $C(10) = C(5)$

$= C(16)$

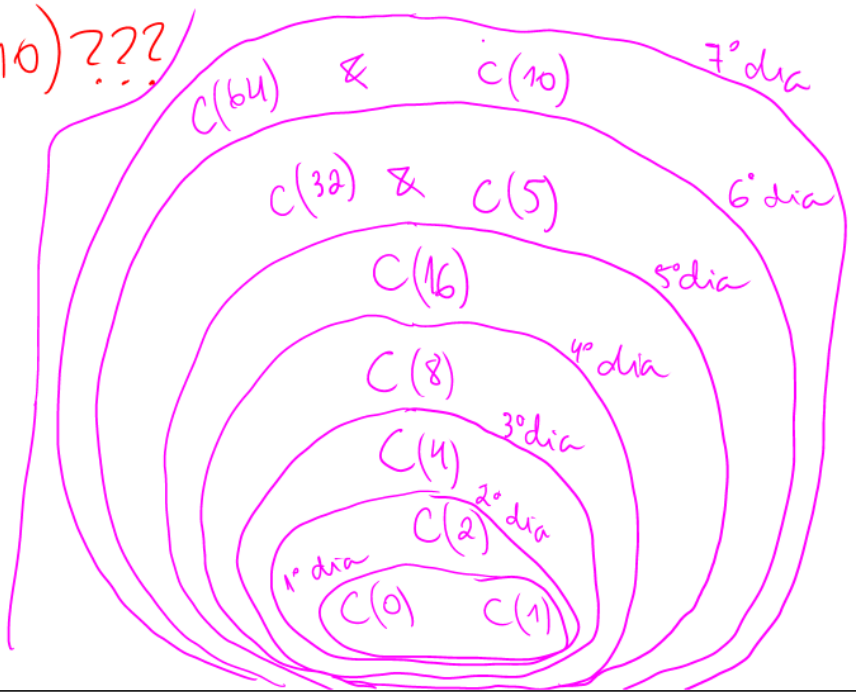
$= C(8)$

$= C(4)$

$= C(2)$

$= C(1)$

$= 1$



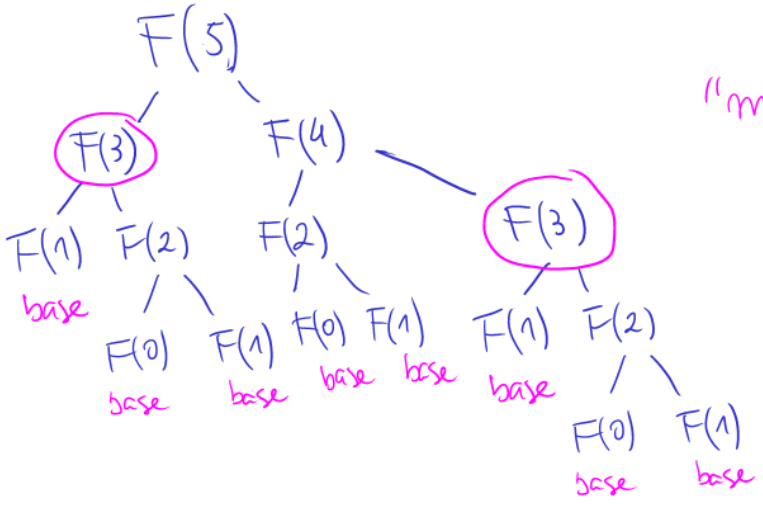
Em matemática, ao fazermos uma nova definição, podemos apenas usar conceitos e objetos que já haviam sido definidos no passado.

Neste sentido, as definições recursivas são um pouco especiais pois parecem depender delas mesmas, o que seria uma dependência circular! Mas para que uma definição recursiva seja "legal",

bem feita, é necessário que se esteja na seguinte situação:

- o objeto que está sendo definido, digamos X, depende de parâmetros [por exemplo, X pode ser uma função e os parâmetros são os possíveis valores de entrada]
- os parâmetros podem ser organizados de forma que a definição de X no caso de parâmetros específicos quaisquer só depende de casos de X com parâmetros que vieram estritamente antes na organização. (dispor os parâmetros em "camadas", como em uma cebola, de forma que a definição de X com os parâmetros em uma camada só dependa de X com parâmetros em camadas estritamente mais internas)

Curiosidade: a "árvore de dependências" de Fibonacci



"memoização"

Dúvidas & Comentários

Suponha verdade: "Isac é o aluno mais alto dessa turma"

Qual a estratégia que isso sugere caso eu queira concluir que um certo alguém tem altura $\leq$ à do Isac

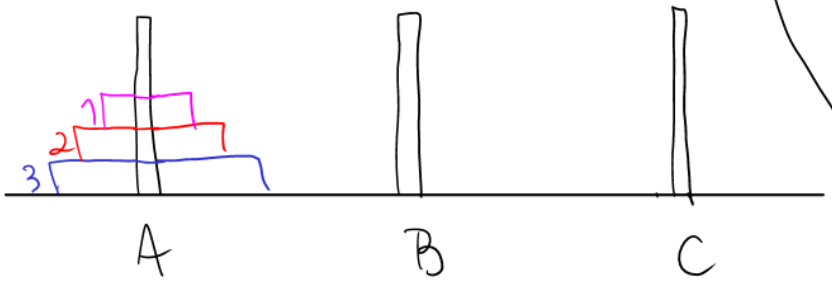
Supondo $a|m$ & $b|m$.

$$m = mmc(a,b) \cdot q + r$$

Suponha, para chegarmos a uma contradição, que $r \neq 0$.

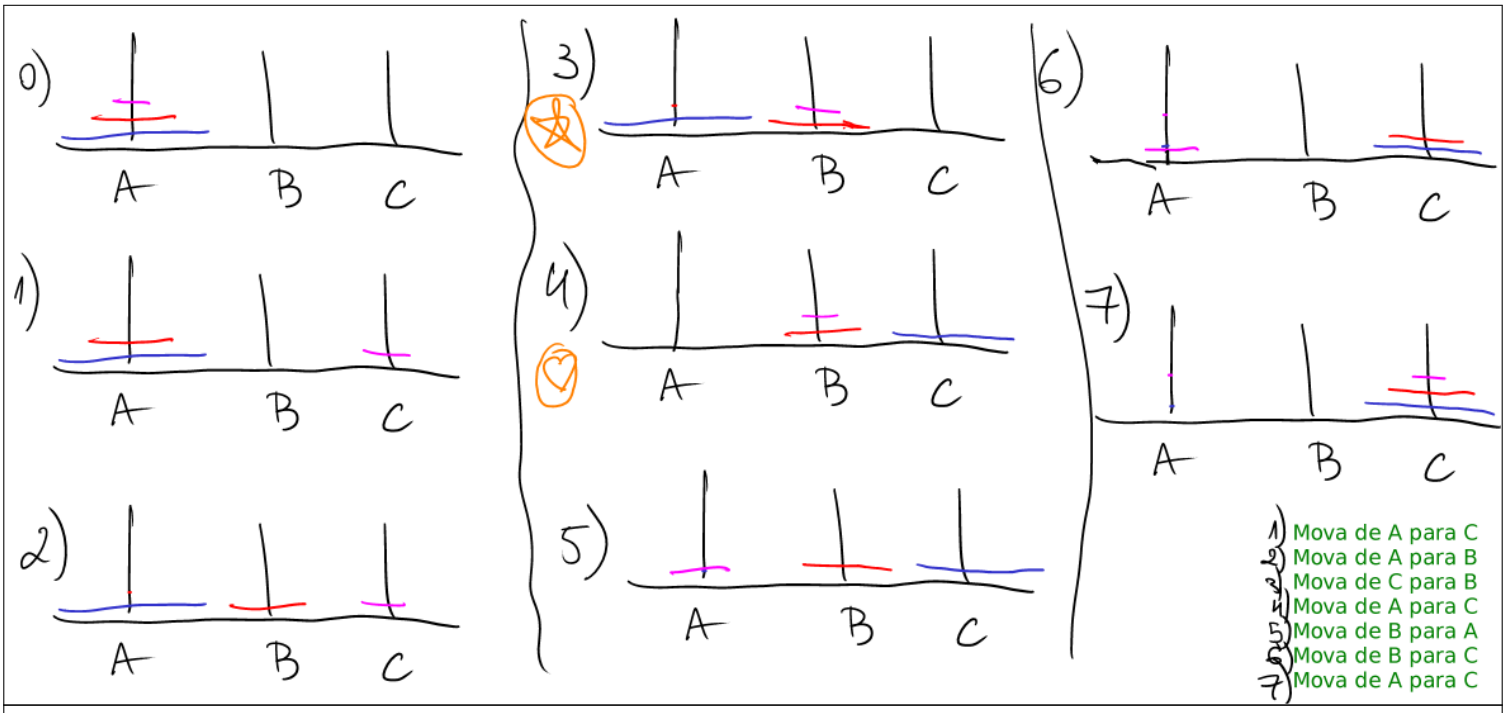
Torres de Hanoi

$n=3$



Objetivo: mover a torre de n discos de A para C respeitando:

1) Apenas o disco no topo de uma torre pode ser movido a cada rodada
2) Um disco não pode ser posto sobre um menor

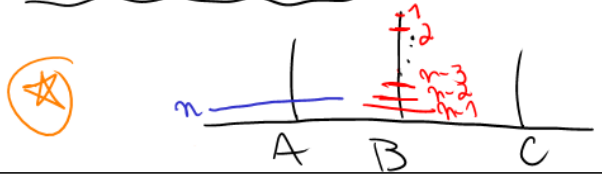


Pensando em uma solução geral.

Caso simples: • $n = 0$ (nenhum movimento!)

• $n = 1$ ("mova o único disco de A para C")

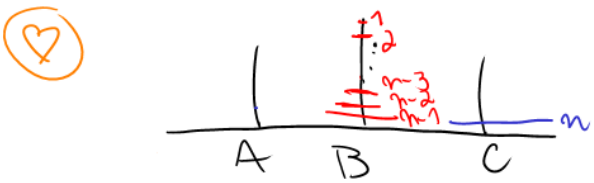
No caso geral ($n > 1$), em algum momento, necessariamente, temos que chegar na situação



pino A: apenas o maior disco
pino B: os outros $n-1$ discos
pino C: vazio

pois esta é a única situação na qual o maior disco pode ser movido para o destino (pino C).

Depois, podemos mover o maior disco de A p/ C



Para chegarmos do início a ⭐ e depois de ♡ ao fim do jogo... recursão!!!

Algoritmo (Torres de Hanoi)

Entrada: n, origem, destino, auxiliar

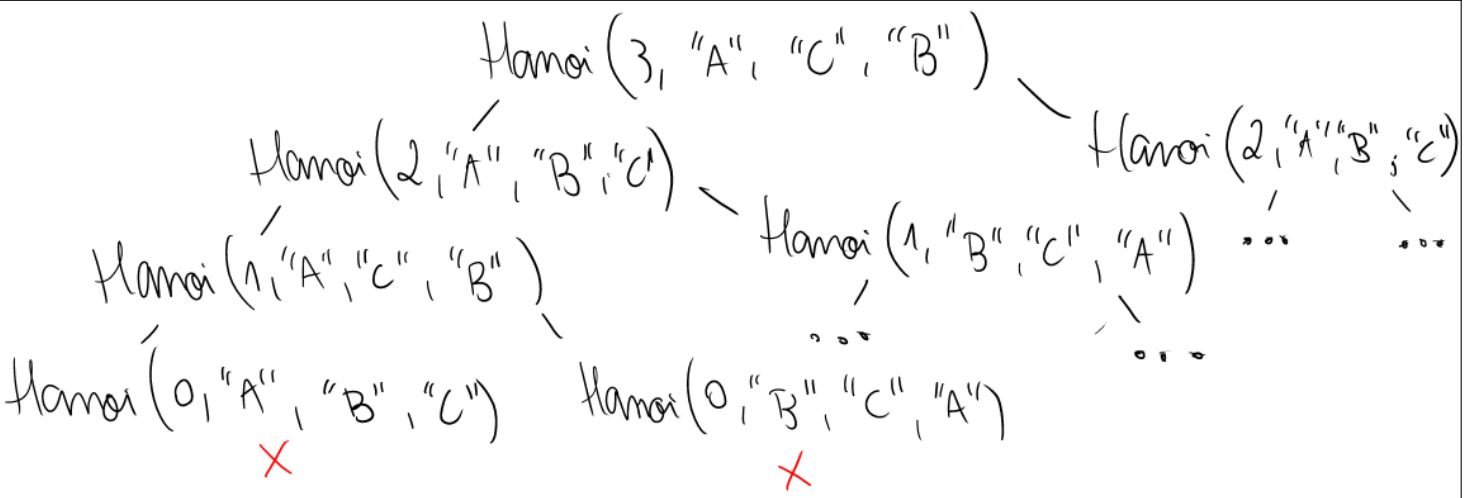
Saída: As instruções de como ganhar o jogo (como mover n discos de origem ao destino)

Passo 1: Se n > 0, fim.
Senão:

Passo 2: Faça Hanoi (n-1, origem, auxiliar, destino) ✳

Passo 3: "Mova de origem para destino" ♡

Passo 4: Faça Hanoi (n-1, auxiliar, destino, origem)



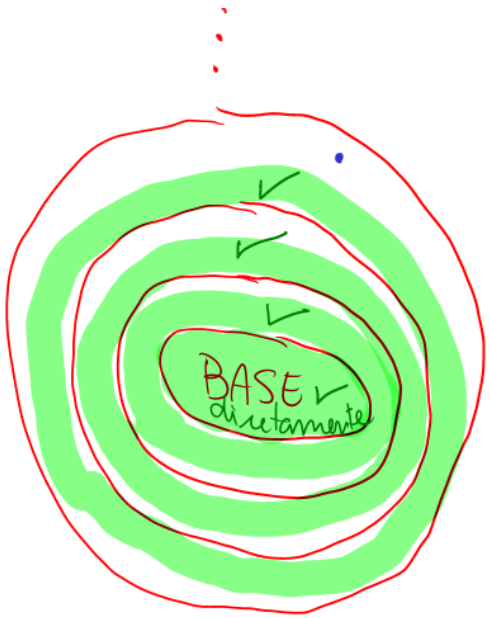
Dúvidas & comentários ?

L3Q4 , L3Q7

Indução

No fundo, uma prova por indução nada mais é do que uma prova construída por recursão: o teorema a ser provado tem vários casos (talvez sejam infinitos!), e vamos ordená-los de forma que:

- há um conjunto de casos que não dependem de outros (casos base da indução). Estes casos devem ser provados diretamente
- (passo de indução) para cada caso que não é da base, este caso depende apenas de casos anteriores na ordem. Aqui vamos supor que os tais casos anteriores já foram provados (Hipótese de Indução, HI), e a tarefa é provar o caso da vez (usando a HI, se for útil)



Uma prova por indução tem 3 partes:

- a organização dos casos (com casos base & de forma que cada caso só dependa de casos anteriores na ordem)
- prova para os casos base (Base da Indução)
- prova do Passo de Indução: dado um caso C fora da base, supondo (HI) que todos os casos anteriores já foram provados, provar o Caso C.

Exemplos

1) Teorema: Para todo $n \in \mathbb{N}$, temos $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$

Prova: Por indução em $n \in \mathbb{N}$, usando a "organização padrão" dos naturais (base é $n=0$, cada n pode depender apenas de casos $k<n$)

Caso Base: $n=0$. Tarefa: mostrar $\sum_{i=0}^{0} i = \frac{0\cdot(0+1)}{2}$

Temos $\sum_{i=0}^{0} i = 0$ & $\frac{0\cdot(0+1)}{2} = 0$, logo são iguais.

Passo: Seja $n > 0$.

(HI) Suponha que $\forall k < n \left(\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \right)$

todos os casos de 0 até $n-1$

Tarefa: mostrar que $\left(\sum_{i=0}^{n} i \right) = \frac{n(n+1)}{2}$

Temos $\sum_{i=0}^{n} i = \left(\sum_{i=0}^{n-1} i \right) + n$

$\sum_{i=0}^{n} i = 0+1+2+\dots+(n-1)+n$

Pela HI com $k = n-1$, temos $\sum_{i=0}^{n-1} i = \frac{(n-1)\cdot(n-1+1)}{2} = \frac{(n-1)\cdot n}{2}$

Portanto $\sum_{i=0}^{n} i = \left(\frac{(n-1)\cdot n}{2}\right) + n$

$$= \frac{(n-1)\cdot n + 2n}{2}$$

$$= \frac{n(n-1+2)}{2} = \frac{n(n+1)}{2}$$

∎

2) Teorema (Binet): Seja $F: \mathbb{N} \to \mathbb{N}$ a função de Fibonacci, ou seja, $F(n) = \begin{cases} n, \text{ se } n \leq 1 \\ F(n-2) + F(n-1), \text{ se } n > 1 \end{cases}$

Então para todo $n \in \mathbb{N}$ temos

$$F(n) = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

(Fazendo $\varphi = \frac{1+\sqrt{5}}{2}$ $\quad \psi = \frac{1-\sqrt{5}}{2}$, fica

$$F(n) = \frac{\varphi^n - \psi^n}{\sqrt{5}}$$

φ: fi

ψ: psi

Prova: Por indução em $n \in \mathbb{N}$

(organização: base 0 & 1, depois disso um número por vez, na ordem natural)

Base: Dois casos, $n=0$ & $n=1$

• $n=0$. Temos $F(0) = 0$, $(\varphi^0 - \psi^0)/\sqrt{5} = (1-1)/\sqrt{5} = 0$, OK!

• $n=1$, Temos $F(1)=1$,

$$\frac{\varphi^1-\psi^1}{\sqrt{5}} = \frac{\left(\frac{1+\sqrt{5}}{2}\right)-\left(\frac{1-\sqrt{5}}{2}\right)}{\sqrt{5}} = \frac{\sqrt{5}}{\sqrt{5}} = 1 \quad \text{ok!}$$

Passo: Seja $n>1$

(HI) $\forall k<n \left(F(k) = \frac{\varphi^k-\psi^k}{\sqrt{5}} \right)$.

Tarefa: provar que $F(n) = \frac{\varphi^n-\psi^n}{\sqrt{5}}$.

Temos $F(n) = F(n-2) + F(n-1)$ pois $n>1$.

Pela HI • para $k=n-2$ temos $F(n-2) = \frac{\varphi^{n-2}-\psi^{n-2}}{\sqrt{5}}$

• para $k=n-1$ temos $F(n-1) = \frac{\varphi^{n-1}-\psi^{n-1}}{\sqrt{5}}$

Logo $F(n) = \frac{\varphi^{n-2}-\psi^{n-2}}{\sqrt{5}} + \frac{\varphi^{n-1}-\psi^{n-1}}{\sqrt{5}}$

$$= \frac{1}{\sqrt{5}}\left[\varphi^{n-2}(1+\varphi) - \psi^{n-2}(1+\psi)\right]$$

Mas $\varphi^2 = \left(\frac{1+\sqrt{5}}{2}\right)^2 = \frac{1+2\sqrt{5}+5}{4} = \frac{3+\sqrt{5}}{2} = \frac{1+\sqrt{5}}{2} + 1$

$= \varphi + 1$

e $\psi^2 = \psi + 1$ analogamente

Logo $F(n) = \frac{1}{\sqrt{5}}\left(\varphi^{n-2}\cdot\varphi^2 - \psi^{n-2}\cdot\psi^2\right) = \frac{\varphi^n-\psi^n}{\sqrt{5}}$ ▨

Na verdade, φ e ψ "nasceram" como soluções da equação $x^2 = 1+x$

Dúvidas & comentários

Ingredientes de uma prova por indução:

- organização dos casos (casos que não dependem de outros [base] & de forma que casos não-base dependam apenas de casos anteriores na organização)
- prova dos casos base
- prova do passo de indução: dado um caso C qualquer fora da base, supondo

Hipótese de Indução (HI): todos os casos anteriores a C já foram provados

provar o caso C.

Análise do Algoritmo das Torres de Hanói

Teorema (Terminação): Para qualquer $n \in \mathbb{N}$ (para qualquer especificação de origem, destino, auxiliar, o algoritmo termina após quantidade finita de passos.)

Prova: Por indução em n.

(organização padrão!)

Caso base: $n = 0$. Neste caso o alg. acaba imediatamente no passo 1.

Passo: Seja $n > 0$. Suponha:

(HI) "Para qualquer $k < n$ e qualquer especificação de origem, destino, auxiliar, o alg. com entradas k, origem, destino, auxiliar termina."

Tarefa: mostrar que "para qualquer especificação de origem,

destino, auxiliar, o alg. com entradas n, origem, destino, auxiliar termina."

Considere uma execução do alg. com entrada n, origem, destino, auxiliar

Como $n > 0$, a execução é Passo 2 → Passo 3 → Passo 4.

- No passo 2 há a chamada recursiva
  Hanoi(n−1, origem, auxiliar, destino)

Como $n-1 < n$, pela HI esse passo termina em algum momento.

- No passo 3 há apenas uma instrução, então termina.
- No passo 4 há a chamada recursiva
  Hanoi(n−1, auxiliar, destino, origem)

Novamente pela HI, esse passo termina

Logo a execução com n discos termina ∎

Teorema (Corretude): Para qualquer $n \in \mathbb{N}$ e para qualquer especificação de origem, destino, auxiliar, a execução do algoritmo com entradas n, origem, destino, auxiliar...
1) respeita as regras do jogo
2) leva a torre de n discos do pino origem ao destino.

Prova: Por indução em n.

(organização padrão)

Caso base: $n=0$. Por vacuidade!

Passo: Seja $n>0$. Suponha
(HI) Para qualquer $k<n$ e para qualquer especificação de origem, destino, auxiliar, a execução do

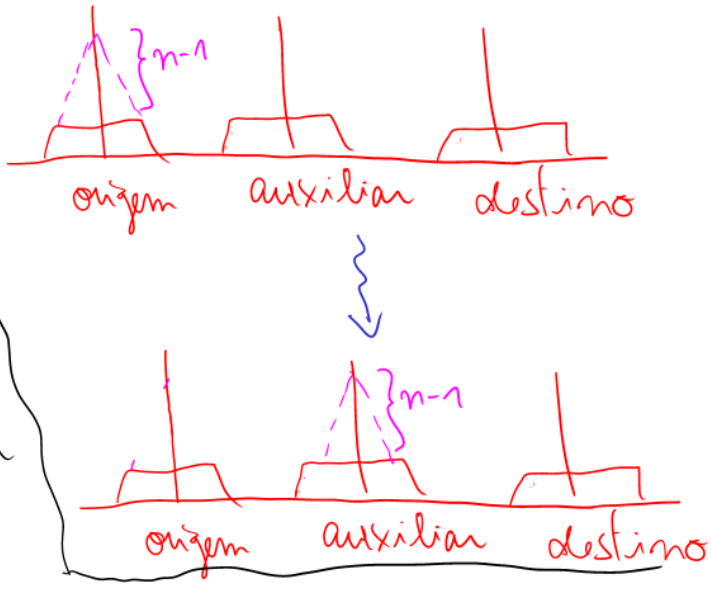
algoritmo com entrada k, origem, destino, auxiliar...
1) respeita as regras do jogo
2) leva a torre de k discos do pino origem ao destino.

Agora considere uma execução do alg com entrada n, origem, destino, auxiliar.
Como $n > 0$, a execução é Passo 2 → Passo 3 → Passo 4.
· No passo 2 há a chamada recursiva

Hanoi (n-1, origem, auxiliar, destino)

Coração da prova: do ponto de vista dos n-1 menores discos, seus movimentos legais "dentro" do jogo "maior" de n discos são exatamente os mesmos que de um jogo onde só existam os n-1 menores discos, pois os movimentos de um disco não são restritos pela posição dos discos maiores.
Portanto, pela HI, os movimentos feitos em Hanoi (n-1, origem, auxiliar, destino) são legais e levam a torre dos n-1 discos no topo do origem para o topo do pino auxiliar:

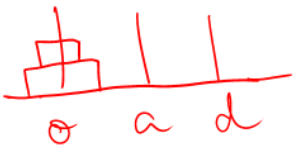
Feito isso, no passo 3 o disco n (agora é o topo do pino origem) pode ser movido ao pino destino (disponível para o disco n, pois os n-1 discos menores do que n estão no pino auxiliar).



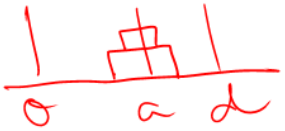
Finalmente, novamente pela HI, no passo 4 a torre de n-1 discos no topo do pino auxiliar é movida para o topo do pino destino respeitando as regras.

Portanto no geral a torre de n discos do topo do pino origem foi levada para o topo do pino destino, sempre respeitando as regras ■

Rascunho o jogo Hanoi(2, origem, auxiliar, destino), literalmente, é partir de



e chegar em

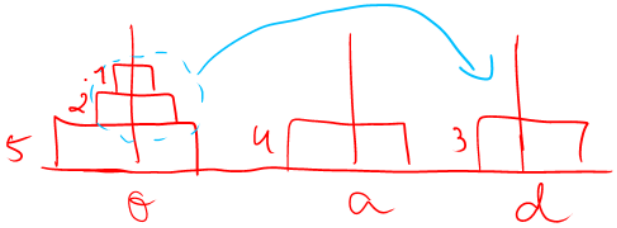


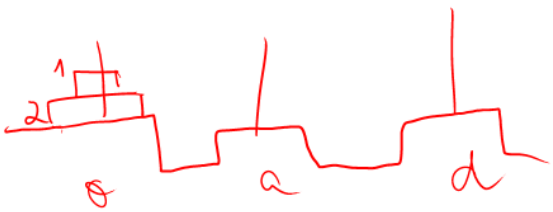
← a HI diz que esse está correto

o que preciso este ↘ é que mas a chamada recursiva de Hanoi(2, origem, auxiliar destino) estejá correto

no passo 2 de Hanoi(3, origem, destino, auxiliar) é partir de

para chegar em





Hanoi(2, o, d, a)

regras respeitadas (pela HI)

Teorema: Para mover n discos, o algoritmo manda fazer $2^n - 1$ movimentos.

Prova: Exercício!

Rascunho: $\left(2^{n-1} - 1\right) + 1 + \left(2^{n-1} - 1\right)$

$= 2^n - 1$

Dúvidas & Comentários

Hanoi

n	quantas linhas imprime
0	0
1	1
2	3
3	7
4	15
5	31
	⋮

Conjectura: $2^n - 1$

↑

fórmula fechada

Se chamarmos de $T(n)$ o número de linhas que são impressas quando rodamos Hanoi com n discos, temos

$$\begin{cases} T(0) = 0 \\ \text{Para } n > 0: \quad T(n) = T(n-1) + 1 + T(n-1) \end{cases}$$

$$= 2 \cdot T(n-1) + 1$$

Relação de recorrência

Teorema: Para todo $n > 0$ temos $T(n) = 2^n - 1$.

Prova: Por indução em n.
(organização padrão)

Caso base: $n = 0$. Temos $T(0) = 0$ & $2^0 - 1 = 0$ também

Passo: Seja $n > 0$. Suponha

(HI) $\forall k < n \; (T(k) = 2^k - 1)$ ← $k = n-1$ é um desses casos

Tarefa: provar $T(n) = 2^n - 1$

Como $n > 0$, temos

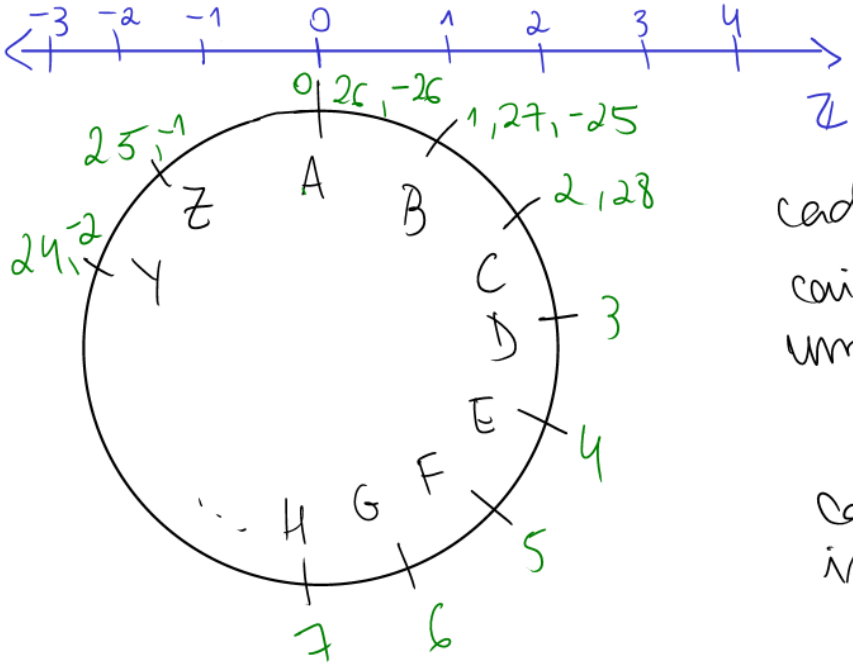
$$T(n) = 2 \cdot T(n-1) + 1$$

$$\overset{HI}{=} 2 \cdot (2^{n-1} - 1) + 1$$

$$= 2^n - 2 + 1$$

$$= 2^n - 1$$ ∎

$n = 26$



cada número cai em exatamente uma casa

&

cada casa recebe infinitos números

Como vimos, <u>para o objetivo de codificar letras</u>, os números que caem numa mesma casa "dão no mesmo", são <u>equivalentes</u>.

Dados inteiros x, y, eles caem na mesma casa sse $x \% 26 = y \% 26$

sse $26 \mid (x - y)$

sse $\exists k \in \mathbb{Z} \; (x = y + k \cdot 26)$

Def: A relação de **congruência módulo n** é definida da seguinte forma.

Dados $x, y \in \mathbb{Z}$, dizemos que **x é congruente a y módulo n**, denotado $x \equiv y \pmod{n}$, quando $\exists k \in \mathbb{Z} \left(x = y + k \cdot n \right)$

[equivalente: $\cdot\ n \mid (x-y)$

$\cdot\ x \% n = y \% n$]

Essa relação é exemplo de uma **relação de equivalência**

Def: Seja R uma relação em um universo U.

1) Dizemos que R é **reflexiva** se todo elemento do universo U está relacionado consigo mesmo pela relação R.

Em símbolos: $\forall x \in U \left(x R x \right)$

Exemplos: $U = \mathbb{Z}$, R = divisibilidade

$U = \mathbb{N}$, R = menor ou igual

U = qualquer conjunto, R = igualdade

2) Dizemos que R é **simétrica** se, para qualquer par de elementos do universo, se eles estão relacionados "em uma direção", então também estão relacionados "na direção oposta"

dá no mesmo vice-versa ↔ (exercício!)

Em símbolos: $\forall x, y \in U \left(x R y \longrightarrow y R x \right)$

Exemplos: U = qualquer conjunto, R = igualdade

U = as pessoas, R = ser irmão

Não exemplo: $U = \mathbb{Z}$, R = divisibilidade (não é simétrica)

3) Dizemos que R é <u>transitiva</u> se

$$\forall x, y, z \in U \left((xRy \wedge yRz) \longrightarrow xRz \right)$$

Graficamente:

<u>Exemplos</u>: $U = \mathbb{Z}$, R = divisibilidade

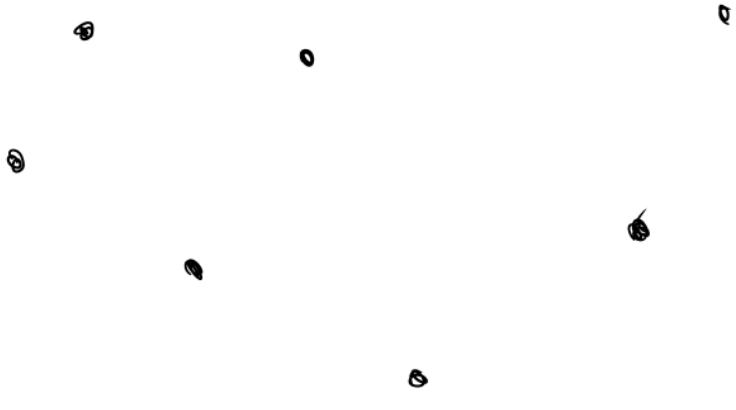
$U = \mathbb{R}$, R = menor que (ou menor ou igual)

U = qualquer conjunto, R = igualdade

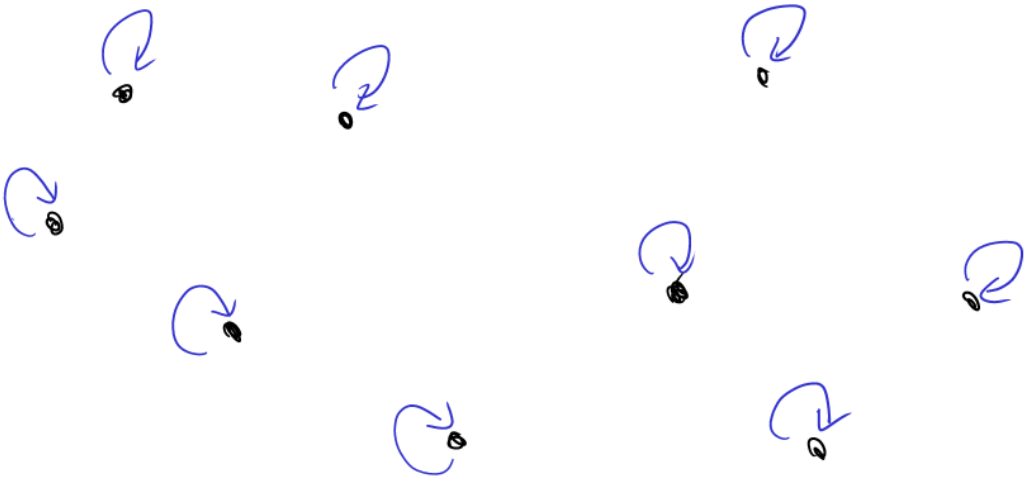
4) Dizemos que R é <u>relação de equivalência</u> se é reflexiva, simétrica e transitiva

---

Representação gráfica de uma relação de equivalência

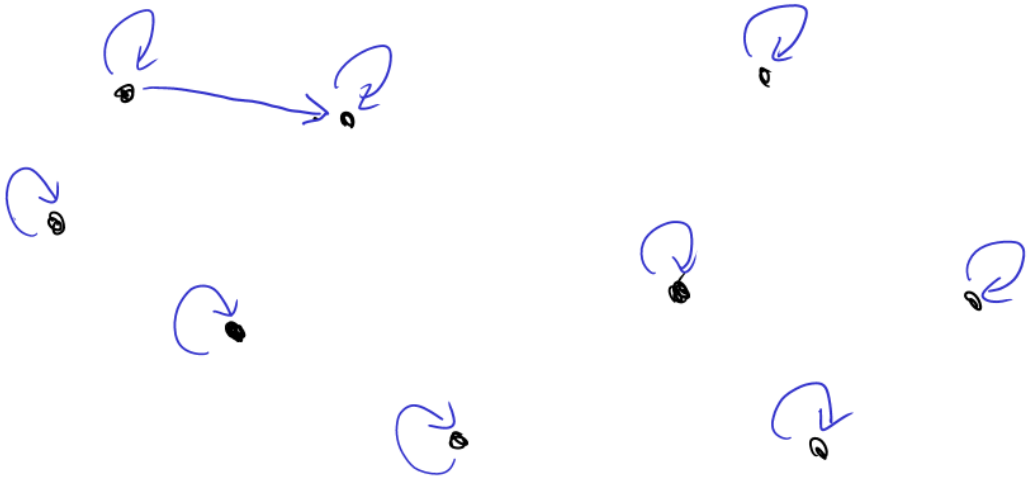


não reflexiva!!
é simétrica
é transitiva

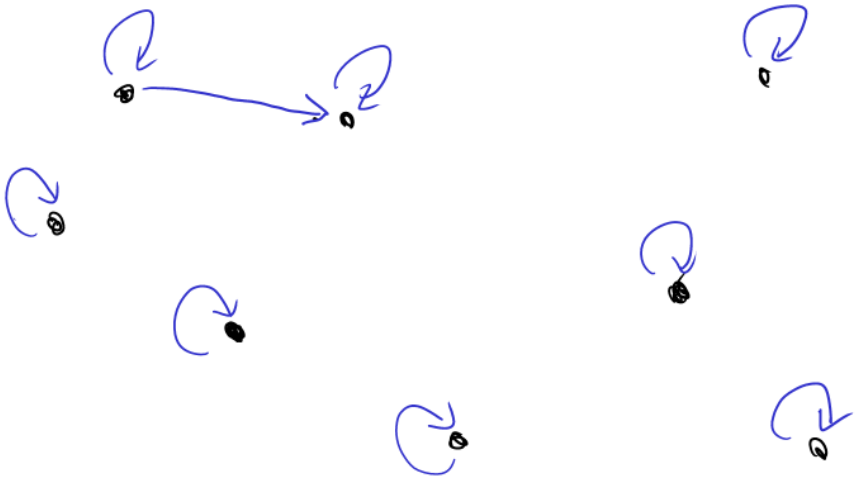


é reflexiva
é simétrica
é transitiva

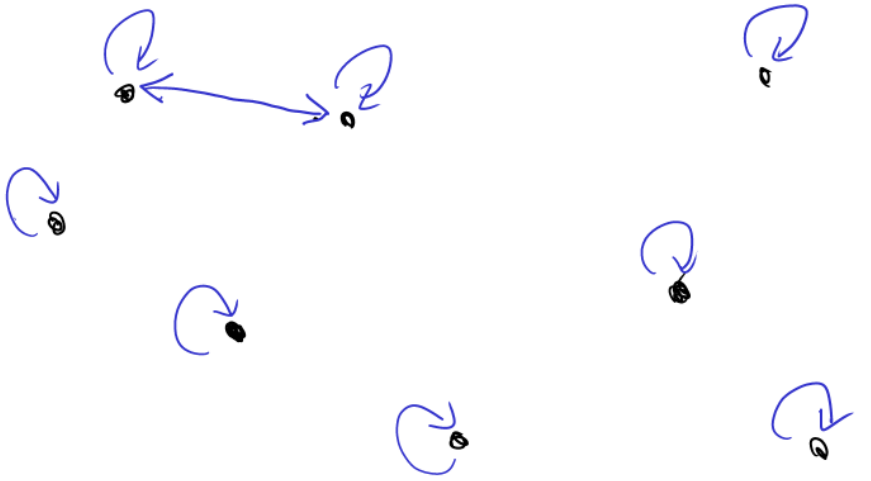
Igualdade



Dúvidas & comentários?

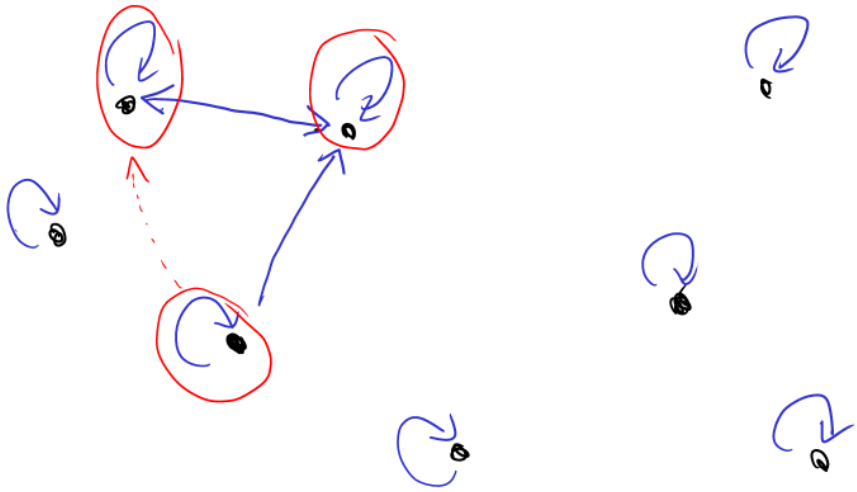


é reflexiva
não é simétrica
é transitiva

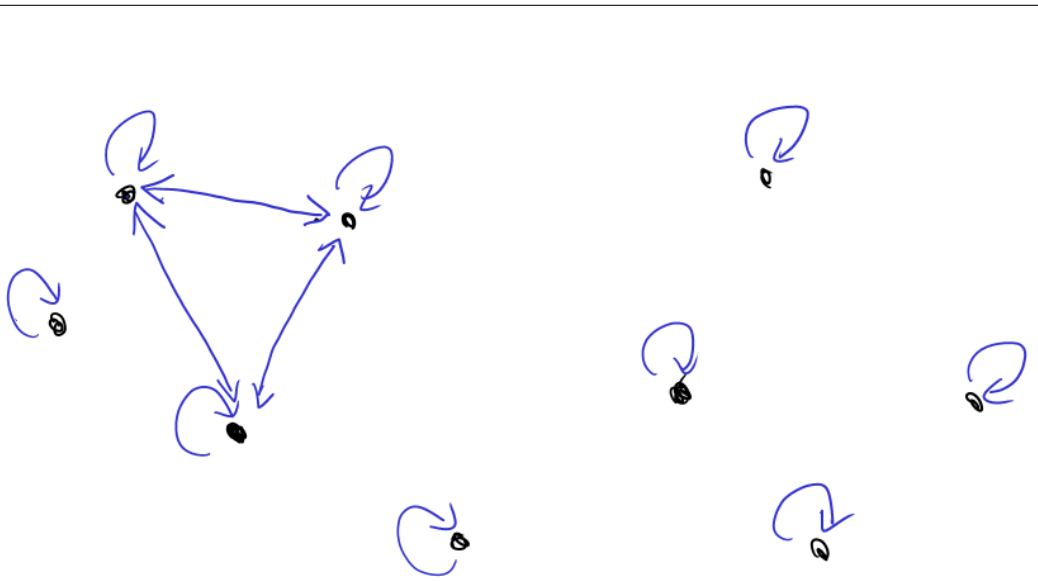


é reflexiva
é simétrica
é transitiva

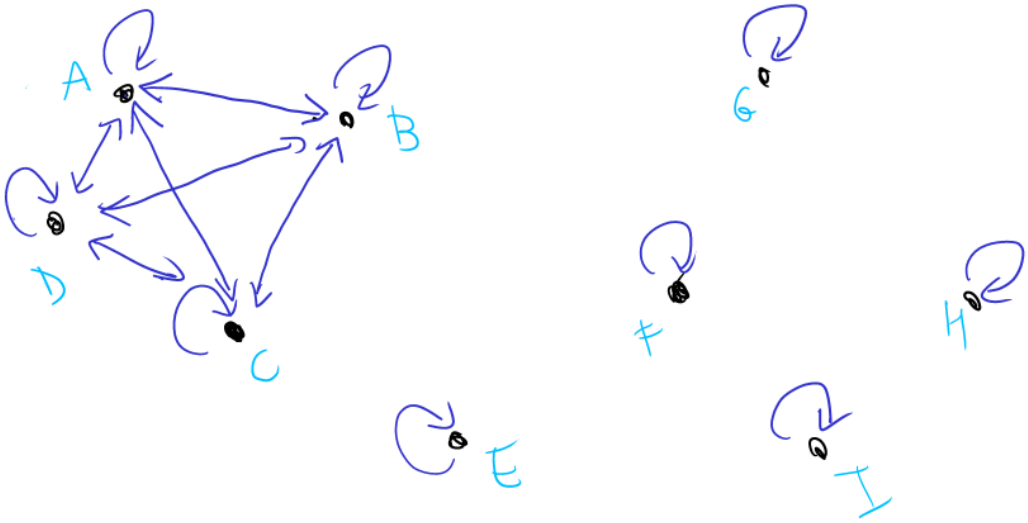
é relação de equivalência



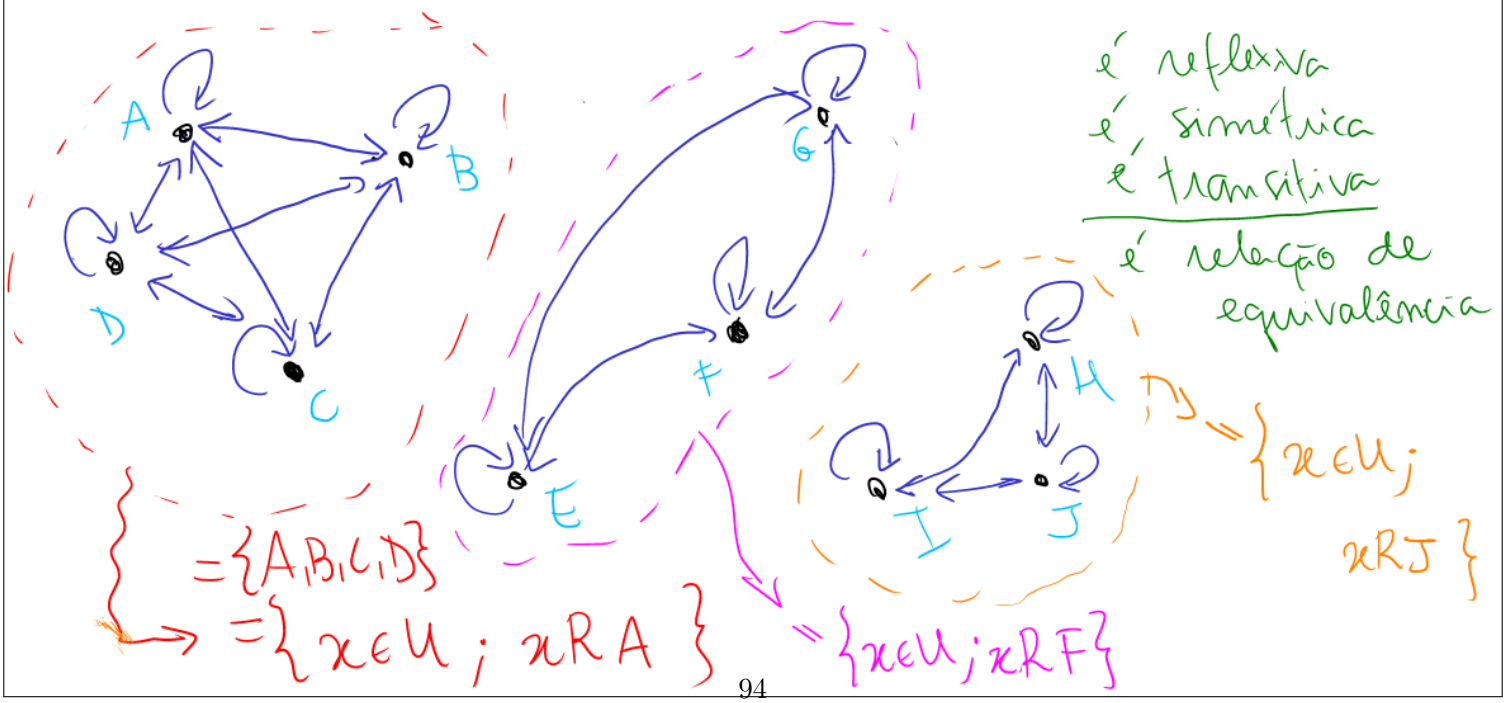
é reflexiva
não é simétrica
não é transitiva



é reflexiva
é simétrica
é transitiva
é relação de equivalência



é reflexiva
é simétrica
é transitiva
é relação de equivalência



é reflexiva
é simétrica
é transitiva
é relação de equivalência

def: Dados um universo U, uma relação de equivalência R em U e um elemento $x \in U$, chamamos de **classe de equivalência de x**, denotado por $\bar{x}$, o conjunto de elementos relacionados a x

$$\bar{x} = \{ y \in U \; ; \; xRy \}$$

x é chamado de **representante** da classe $\bar{x}$.

Teorema: Dados um universo U e uma relação de equivalência R em U.

Para todos $x, y \in U$, temos

$$xRy \quad \text{sse} \quad \bar{x} = \bar{y}.$$

"qualquer elemento de uma classe é representante daquela classe"

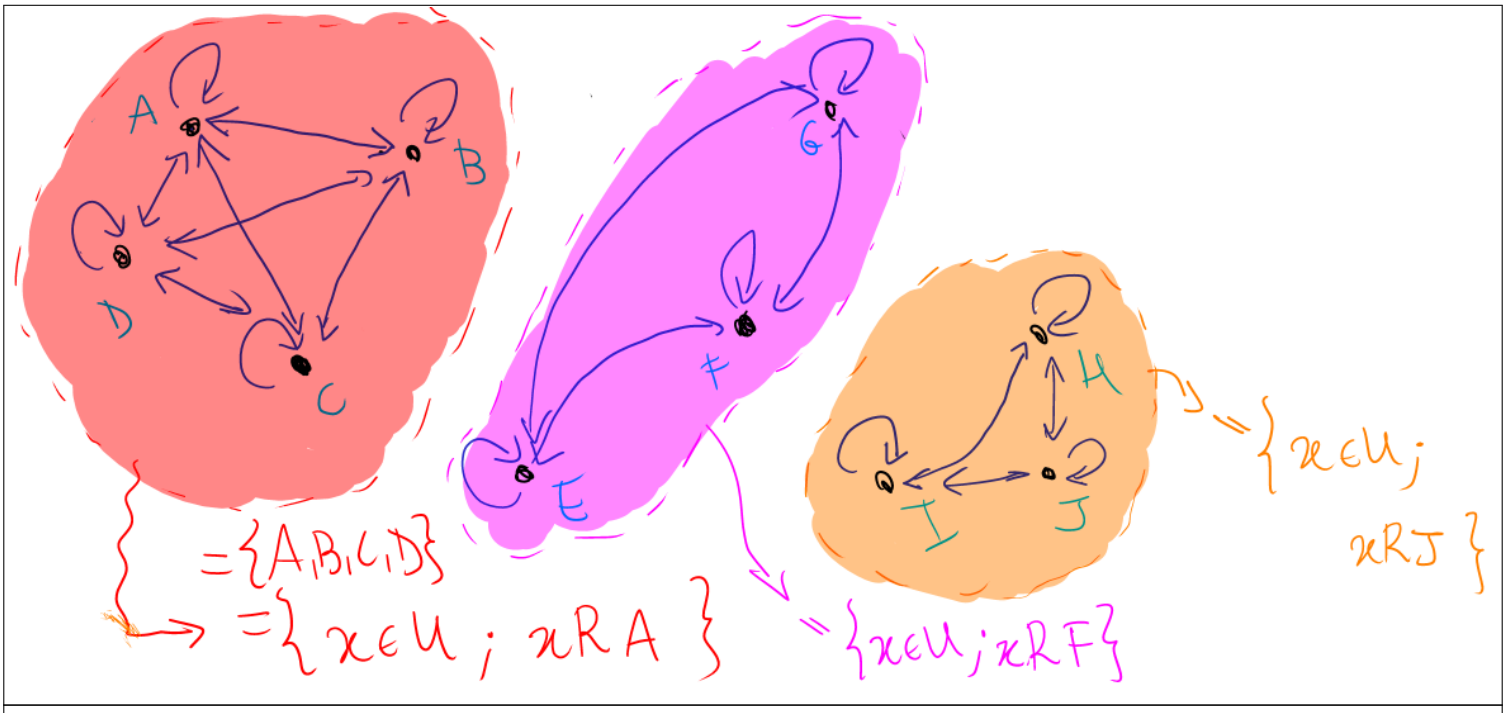
Prova: Exercício.

---

def: Dados um universo U, uma relação de equivalência R em U. Chamamos de **quociente de U por R**, denotado U/R,

o conjunto das classes de equivalência dos elementos de U pela relação R.

$$U/R = \{ \bar{x} \; ; \; x \in U \}$$



Teorema: Para qualquer $n \in \mathbb{Z}$ com $n \neq 0$, a relação de congruência módulo n é uma relação de equivalência

Prova: exercício.

def: Seja $n \in \mathbb{Z}$ com $n \neq 0$

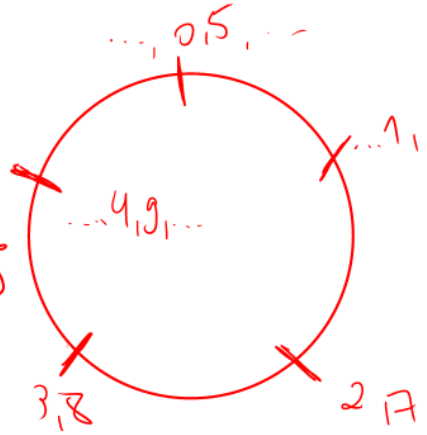
O quociente de $\mathbb{Z}$ pela relação de congruência módulo n é chamado de "os inteiros módulo n", denotado $\mathbb{Z}_n$

Note que $\mathbb{Z}_n$ tem n elementos (exercício)

Exemplo: A relação de congruência módulo 5

$\mathbb{Z}_5 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}\}$

$= \{\overline{15}, \overline{-4}, \overline{22}, \overline{108}, \overline{4}\}$



dado $x \in \mathbb{Z}$, temos

$\overline{x} = \{y \in \mathbb{Z} ; x \equiv y \bmod 5\}$

$= \{y \in \mathbb{Z} ; \exists k \in \mathbb{Z} \ (x = y + 5 \cdot k)\}$

$\overline{0} = \{\ldots, -10, -5, 0, 5, 10, \ldots\}$

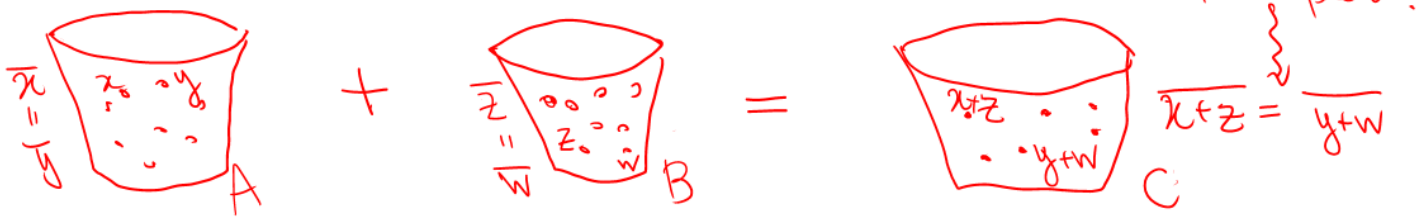
$\overline{1} = \{\ldots, -9, -4, 1, 6, 11, \ldots\}$

Aritmética modular (aritmética em $\mathbb{Z}_n$)

Seja $n \in \mathbb{Z}$, $n \neq 0$, e consideremos $\mathbb{Z}_n$

Queremos definir aritmética ($+, -, \times, \div$, etc) em $\mathbb{Z}_n$.

Rascunho



Dúvidas & Comentários

Na aula de hoje, considere um $n \in \mathbb{Z}$, $n \neq 0$, fixo.

Teorema: Em $\mathbb{Z}_n$, para quaisquer $x, y, z, w \in \mathbb{Z}$,

se $\overline{x} = \overline{y}$ e $\overline{z} = \overline{w}$

então $\overline{x+z} = \overline{y+w}$

Alternativa
Se $x \equiv y \bmod n$
e $z \equiv w \bmod n$
então $x+z \equiv y+w \bmod n$

Prova: Suponha $x \equiv y \bmod n$ & $z \equiv w \bmod n$

Logo $\exists k, l \in \mathbb{Z}$ $(x - y = k \cdot n$ & $z - w = l \cdot n)$

Quero mostrar: $\exists m \in \mathbb{Z}$ $\big(\underbrace{(x+z) - (y+w)}_{x+z-y-w} = m \cdot n \big)$

Portanto $x - y + z - w = k \cdot n + l \cdot n$

ou seja $(x+z) - (y+w) = (k+l) \cdot n$

Logo $x+z \equiv y+w \bmod n$ (pois $k+l \in \mathbb{Z}$) ▨

Def (Adição em $\mathbb{Z}_n$). Em $\mathbb{Z}_n$, definimos $\forall a, b \in \mathbb{Z}$

$$\overline{a} + \overline{b} = \overline{a+b}$$

↑ essa definição só faz sentido por causa do teorema acima

Exemplo: em $\mathbb{Z}_5$

$\overline{12345678} + \overline{12345678} = \overline{4} + \overline{4}$

$\overline{11} + \overline{11} = \overline{-2} = \overline{3} = \overline{8}$

- Subtração em $\mathbb{Z}_n$

Dados $a, b \in \mathbb{Z}$, queremos definir $\overline{a} - \overline{b}$

Na verdade queremos definir $-\overline{b}$, ou seja, quem é $\overline{x} \in \mathbb{Z}_n$ tal que $\overline{b} + \overline{x} = \overline{0}$

Logo $x = -b$ serve!

Teorema: Em $\mathbb{Z}_n$, se $\bar{x} = \bar{y}$ então $-\bar{x} = -\bar{y}$ (Exercício)

def: • Dado $a \in \mathbb{Z}$, em $\mathbb{Z}_n$ definimos $-\bar{a} = \overline{-a}$

↑ justificado pelo Teorema acima

• Dados $a, b \in \mathbb{Z}$, em $\mathbb{Z}_n$ definimos

$$\bar{a} - \bar{b} = \bar{a} + (-\bar{b}) = \overline{a-b}$$

• Multiplicação em $\mathbb{Z}_n$

Quero $\bar{a} \cdot \bar{b} = \overline{a \cdot b}$

Teorema: Em $\mathbb{Z}_n$, para quaisquer $x, y, z, w \in \mathbb{Z}$,

se $\bar{x} = \bar{y}$ e $\bar{z} = \bar{w}$

então $\overline{x \cdot z} = \overline{y \cdot w}$

Alternativa

se $x \equiv y \bmod n$ e $z \equiv w \bmod n$ então $x \cdot z \equiv y \cdot w \bmod n$

Prova: Suponha $x \equiv y \bmod n$ & $z \equiv w \bmod n$

Então $\exists k, l \in \mathbb{Z} (x - y = k \cdot n \ \& \ z - w = l \cdot n)$

Quero: $\exists m \in \mathbb{Z} (xz - yw = m \cdot n)$

Tentativa 1: $(x-y)(z-w) = (k \cdot n)(l \cdot n)$

$xz - xw - yz + yw = (kln)n$

... não deu

Temos $x = y + kn$ & $z = w + ln$

Logo $xz = yw + yln + zkn + kln^2$

$(xz - yw) = (yl + zk + kln)n$ $m \in \mathbb{Z}$

Logo $xz \equiv yw \bmod n$. ▨

def: Dados $a, b \in \mathbb{Z}$, em $\mathbb{Z}_n$ definimos

$$\overline{a} \cdot \overline{b} = \overline{a \cdot b}$$

Exemplo: Em $\mathbb{Z}_3$, $\overline{12345678} \cdot \overline{12345689} = \overline{0}$.

- Divisão em $\mathbb{Z}_n$.

"Quase" não temos $\div$ em $\mathbb{Z}$

Queremos $\overline{a} \div \overline{b}$. Na verdade "$\div \overline{b}$" é

multiplicar pelo inverso multiplicativo de $\overline{b}$, ou seja, multiplicar pelo $\overline{x} \in \mathbb{Z}_n$ tal que

$\overline{b} \cdot \overline{x} = \overline{b \cdot x} = \overline{1}$

Portanto queremos saber quais são os $b \in \mathbb{Z}$ para os quais $\exists x \in \mathbb{Z}$ tal que $\overline{bx} = \overline{1}$

ou seja $\exists x \in \mathbb{Z}$ tal que $bx \equiv 1 \mod n$

ou seja $\exists x \in \mathbb{Z}\ \exists k \in \mathbb{Z}$ tais que $bx = 1 + k \cdot n$

ou seja $\exists x \in \mathbb{Z}\ \exists y \in \mathbb{Z} (b \cdot x + n \cdot y = 1)$ Bézout!!

Pelo que visto acima: $\overline{b}$ tem inverso em $\mathbb{Z}_n$ $\Leftarrow$ "b tem inverso módulo n"

sse

$$\exists x, y \in \mathbb{Z} \left(b \cdot x + n \cdot y = 1 \right)$$

Teorema: Dado $b \in \mathbb{Z}$, temos

$$\exists x, y \in \mathbb{Z} \left(b \cdot x + n \cdot y = 1 \right)$$

sse

$$mdc(b, n) = 1$$

Prova ($\Leftarrow$) Teorema de Bézout!

($\Rightarrow$) Sejam $x, y \in \mathbb{Z}$

tais que $bx + ny = 1$

Como $mdc(b,n) \mid b$ & $mdc(b,n) \mid n$,

temos (Lista 2) $mdc(b,n) \mid (bx + ny)$

logo $mdc(b,n) \mid 1$.

Portanto $mdc(b,n) = 1$ ∎

Teorema (Teorema dos Inversos em $\mathbb{Z}_n$). Dado $b \in \mathbb{Z}$, temos que $\overline{b}$ tem inverso em $\mathbb{Z}_n$ sse $mdc(b,n) = 1$.

(Quando ele existe, o inverso de $\overline{b}$ pode ser encontrado pelo Alg. de Euclides Estendido)

def: Dados $a, b \in \mathbb{Z}$ com $mdc(b,n) = 1$, definimos $\overline{a} \div \overline{b} = \overline{a} \cdot \overline{x}$

onde $\overline{x}$ é o inverso multiplicativo de $\overline{b}$ em $\mathbb{Z}_n$.

Notação: Denotamos o inverso multiplicativo de $\overline{b}$ por $\frac{\overline{1}}{\overline{b}}$ ou $(\overline{b})^{-1}$

Exemplo: • Em $\mathbb{Z}_5$, $\overline{1}, \overline{2}, \overline{3}, \overline{4}$ têm inversos multipl.

$(\overline{1})^{-1} = \overline{1}$, $(\overline{2})^{-1} = \overline{3}$, $(\overline{3})^{-1} = \overline{2}$, $(\overline{4})^{-1} = \overline{4}$

Portanto, em $\mathbb{Z}_5$, temos $\overline{123} \div \overline{2} = \overline{3} \cdot \overline{3}$

$= \overline{4}$

• Em $\mathbb{Z}_8$, $\overline{1}, \overline{3}, \overline{5}, \overline{7}$

$(\overline{1})^{-1} = \overline{1}$, $(\overline{3})^{-1} = \overline{3}$, $(\overline{5})^{-1} = \overline{5}$, $(\overline{7})^{-1} = \overline{7}$

Portanto $\overline{6} \div \overline{7} = \overline{6} \cdot \overline{7} = \overline{2}$ em $\mathbb{Z}_8$.

- Potenciação em $\mathbb{Z}_n$

Tentativa: $\overline{a}^{\overline{b}} = \overline{(a^b)}$ se $b>0$??

Precisaríamos provar: $\forall x y z w \left(\overline{x}=\overline{y} \wedge \overline{z}=\overline{w}\right) \longrightarrow \left(\overline{x^z} = \overline{y^w}\right)$

Mas isso não é verdade!

Contraexemplo: $n=3$, $x=2$, $y=2$, $z=3$, $w=6$

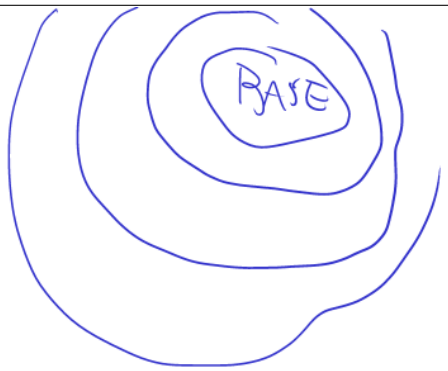
Então $\overline{x^z} = \overline{8} = \overline{2}$, mas $\overline{y^w} = \overline{64} = \overline{1}$, logo $\overline{x^z} \neq \overline{y^w}$

Logo não definiremos a operação $\overline{a}^{\overline{b}}$

Mas podemos definir $\overline{a}^k$ com $k \in \mathbb{N}$

("multiplicação repetida")

Dúvidas & Comentários



$$F: \mathbb{N} \to \mathbb{N}$$

$$F(n) = \begin{cases} n & , n \leq 1 \\ F(n-2) + F(n-1) & , \text{C. C.} \end{cases}$$

$$\forall n \in \mathbb{N}: \quad F(n) = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

Ex: $\overline{10} \div \overline{30} = \overline{10} \cdot \overline{2}$ em $\mathbb{Z}_{59}$

$= \overline{20}$

$\overline{10} \div \overline{3} = \overline{10} \cdot \overline{20}$ em $\mathbb{Z}_{59}$

$= \overline{200}$

$= \overline{23}$

Seja $n \neq 0$, $n \in \mathbb{Z}$, e consideremos $\mathbb{Z}_n$

def: Dado $\bar{a} \in \mathbb{Z}_n$, definimos $\bar{a}^k$ com $k \in \mathbb{N}$:

$$\bar{a}^k = \underbrace{\bar{a} \cdot \bar{a} \cdot \bar{a} \cdot \cdots \cdot \bar{a}}_{k \text{ termos } \bar{a}}$$

ou, por recursão,

$$\bar{a}^k = \begin{cases} \bar{1} & , \text{se } k = 0 \\ \bar{a} \cdot \bar{a}^{k-1} & , \text{se } k > 0 \end{cases}$$

Teorema: $\overline{a}^k = \overline{a^k}$

Prova: Exercício!

Def: Dado $x \in \mathbb{Z}$, a forma reduzida de $x \bmod n$ é o $y \in \mathbb{Z}$ satisfazendo

1) $x \equiv y \bmod n$
2) $0 \leq y < n$

(y é o resto da divisão de x por n)

Exemplos: Calcule a forma reduzida de

a) $2^{12} \bmod 31$

$2^1 \equiv 2 \bmod 31$

$2^2 \equiv 4 \bmod 31$

$2^3 \equiv 8 \bmod 31$

$(2^2)^2 \equiv 4^2 \bmod 31$

$2^4 \equiv 16 \bmod 31$

$2^5 \equiv 2 \cdot 16 \equiv 1 \bmod 31$

Eureka!!

$2^6 \equiv 2 \bmod 31$

$2^7 \equiv 4 \bmod 31$

$2^8 \equiv 8 \bmod 31$

$2^9 \equiv 16 \bmod 31$

$2^{10} \equiv 1 \bmod 31$

$2^{11} \equiv 2 \bmod 31$

$2^{12} \equiv 4 \bmod 31$

$\vdots$

$2^{513} \equiv 8 \bmod 31$

$\hookrightarrow \overline{2}^{513} = \overline{8}$ em $\mathbb{Z}_{31}$

$2^1 \equiv 2 \bmod 5 \equiv 2^5 \bmod 5 \equiv 2^{513} \bmod 5$

$2^2 \equiv 4 \bmod 5 \equiv 2^6 \bmod 5$

$2^3 \equiv 3 \bmod 5 \equiv 2^7 \bmod 5$

$2^4 \equiv 1 \bmod 5 \equiv 2^8 \bmod 5$

b) $10^{135} \mod 7$

$10^1 \equiv 3 \mod 7$

$10^2 \equiv 2 \mod 7$

$10^3 \equiv 6 \mod 7$

$10^4 \equiv 4 \mod 7$

$10^5 \equiv 5 \mod 7$

$10^6 \equiv 1 \mod 7$

$10^{135} \equiv 10^3 \equiv 6 \mod 7$ //

$\Vert\Vert$

$(10^6)^{22} \cdot 10^3 \equiv 1^{22} \cdot 10^3 \equiv 10^3$

c) $6^{35} \mod 16$

$6^1 \equiv 6 \mod 16$

$6^2 \equiv 4 \mod 16$

$6^3 = 8 \mod 16$

$6^4 \equiv 0 \mod 16$

!!! Eureka

$6^{35} \equiv 0 \mod 16$

$6^k \equiv 0 \quad \forall k \geq 4$!!!

d) $3^{123} \mod 24$

$3^1 \equiv 3 \mod 24 \equiv 3^3 \equiv 3^5 \equiv 3^7 \equiv \ldots \equiv 3^{123}$

$3^2 \equiv 9 \mod 24 \equiv 3^4 \equiv 3^6 \equiv 3^8$

e) $3^{64} \mod 31$

$3^1 \equiv 3 \mod 31$

$3^2 \equiv 9 \mod 31$

$3^3 \equiv 27 \mod 31$

$2^5 \equiv 1 \mod 31$

$3^4 \equiv 19 \mod 31$

$3^5 \equiv 26 \mod 31$

$3^6 \equiv 16 \mod 31$

$\| \|$

$2^4 \mod 31$

$3^6 \equiv 2^4$

$3^{64} \equiv (3^6)^{10} \cdot 3^4 \equiv (2^4)^{10} \cdot 19$

$\equiv 1^{10} \cdot 19$

$\equiv 19 \mod 31$

Dívida do passado: o nome "congruência"

---

quer dizer que somar equivalentes resulta em equivalentes, e igualmente para multiplicação.

— × —

O Pequeno Teorema de Fermat

Teorema (PTF, 2ª versão, PTF2) Sejam $p \in \mathbb{N}$ e $a \in \mathbb{Z}$ com $\text{mdc}(a,p)=1$.

Se p é primo então $a^{p-1} \equiv 1 \mod p$ [$\bar{a}^{p-1} = \bar{1}$ em $\mathbb{Z}_p$]

Teorema (PTF, 1ª versão, PTF1). Sejam $p \in \mathbb{N}, a \in \mathbb{Z}$.

Se p é primo então $a^p \equiv a \mod p$

Instigação: suponha que tenhamos $n \in \mathbb{N}$ e $a \in \mathbb{Z}$ com $\text{mdc}(a,n)=1$

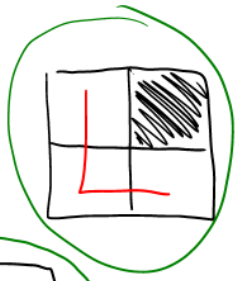
Se calculamos $a^{n-1} \mod n$ e o resultado não é 1, então... ?

Dúvidas & comentários

- Semana SIAc (14, 15, 17 de fevereiro)
  - Com aulas assíncronas
  - Sem aulas ao vivo

L4Q8

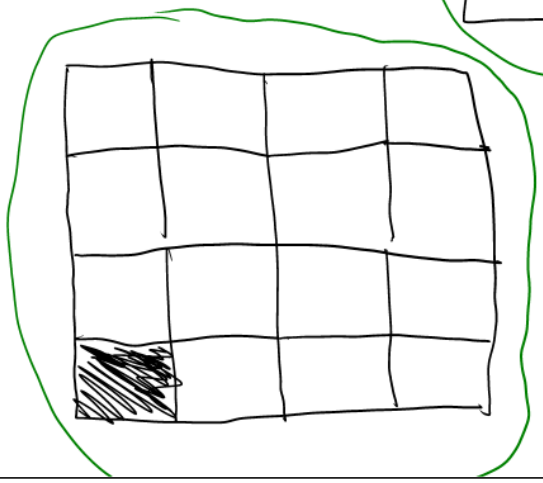
exemplo $n = 1$



$n = 0$??

$n = 2$

?



$$\nexists x, y \in \mathbb{Z} \left(x \neq 0 \wedge y \neq 0 \wedge x \cdot y = 0 \right)$$

$$\exists x, y \in \mathbb{Z}_6 \left(x \neq \overline{0} \wedge y \neq \overline{0} \wedge x \cdot y = \overline{0} \right)$$

$\overline{0} = \overline{6}$

- $x = \overline{6}$ não serve pois nesse caso teríamos $x = \overline{0}$
- $x = \overline{2}$, $y = \overline{3}$ servem

a conta $\overline{2} \div \overline{5}$ faz sentido em $\mathbb{Z}_6$

Teorema (PTF, 1ª versão, PTF1). Sejam $p \in \mathbb{N}, a \in \mathbb{Z}$.
se p é primo então $a^p \equiv a \bmod p$
$[\bar{a}^p = \bar{a}$ em $\mathbb{Z}_p]$

Teorema (PTF, 2ª versão, PTF2) Sejam $p \in \mathbb{N}$ e $a \in \mathbb{Z}$ com $mdc(a,p)=1$.
se p é primo então $a^{p-1} \equiv 1 \bmod p$ $[\bar{a}^{p-1} = \bar{1}$ em $\mathbb{Z}_p]$

—x—

Teorema: PTF1 é equivalente a PTF2.

Prova: (PTF1 $\Rightarrow$ PTF2) Suponhamos

$\forall p \in \mathbb{N} \forall a \in \mathbb{Z} \left(p \text{ primo} \Rightarrow a^p \equiv a \bmod p\right)$

Quero provar o PTF2:

$\forall q \in \mathbb{N} \forall b \in \mathbb{Z} \left(mdc(b,q)=1 \Rightarrow \left(q \text{ primo} \Rightarrow b^{q-1} \equiv 1 \bmod q\right)\right)$

Sejam $q \in \mathbb{N}$, $b \in \mathbb{Z}$ tais que $mdc(b,q)=1$

Quero provar: q primo $\Rightarrow b^{q-1} \equiv 1 \bmod q$

Suponha que q seja primo

Quero provar: $b^{q-1} \equiv 1 \bmod q$.

Pelo PTF1, temos $b^q \equiv b \bmod q$. (*)

Como $mdc(b,q)=1$, b tem inverso multiplicativo b^{-1} em módulo q.

Multiplicando ambos os lados de $(\star)$ por b^{-1}, obtemos

$$b^{-1}\cdot b^q \equiv b^{-1}\cdot b \mod q$$

$$b^{-1}\cdot(\underbrace{b\cdot b\cdot b\cdots b}_{q\ b's}) \equiv 1 \mod q$$

$$b^{q-1} \equiv 1 \mod q.$$

ou seja $b^{q-1}\equiv 1 \mod q$. $\square(\Rightarrow)$

(PTF2 $\Rightarrow$ PTF1). Suponhamos o PTF2:

$$\forall q\in\mathbb{N}\ \forall b\in\mathbb{Z}\left(mdc(b,q)=1 \Rightarrow \left(q \text{ primo} \Rightarrow b^{q-1}\equiv 1 \mod q\right)\right)$$

Quero provar o PTF1:

$$\forall p\in\mathbb{N}\ \forall a\in\mathbb{Z}\ (p \text{ primo} \Rightarrow a^p\equiv a \mod p)$$

Sejam $p\in\mathbb{N}$ e $a\in\mathbb{Z}$.

Suponhamos p primo.

Quero provar: $a^p\equiv a \mod p$

Vamos separar em dois casos

Caso 1: Se $mdc(a,p)=1$, então pelo PTF2 temos $a^{p-1}\equiv 1 \mod p$

Logo $a^p\equiv a \mod p$ (mult. por a em ambos os lados)

Caso 2: Se $mdc(a,p)\neq 1$, como p é primo, temos $mdc(a,p)=p$, logo $p\mid a$, portanto $a\equiv 0 \mod p$, e assim também $a^p\equiv 0 \mod p$ ∎

Prova do PTF1: Primeiramente, note que basta provar o que se afirma para $a \in \mathbb{N}$, pois todo inteiro é congruente (mod p) a algum natural.

Então provaremos: $\forall p \in \mathbb{N}\ \forall a \in \mathbb{N}\ \left(p \text{ primo} \Rightarrow a^p \equiv a \mod p\right)$

Seja $p \in \mathbb{N}$.

Vamos provar $\forall a \in \mathbb{N}\ (p \text{ primo} \Rightarrow a^p \equiv a \mod p)$

por indução em a!

A organização dos casos é a usual: base $a=0$, cada $a>0$ vem depois dos $k<a$.

Caso base: $a = 0$. Então $a^p \equiv 0^p \equiv 0 \equiv a \mod p$.

Passo: Seja $a > 0$.

Suponha

(HI) $\forall k < a\ (p \text{ primo} \Rightarrow k^p \equiv k \mod p)$

Quero provar: $p \text{ primo} \Rightarrow a^p \equiv a \mod p$.

Suponha p primo e vamos provar $a^p \equiv a \mod p$.

Como $a > 0$, temos $a = b + 1$ para algum $b \in \mathbb{N}$

Pela HI, temos $b^p \equiv b \mod p$

Quero $(b+1)^p \equiv b+1 \mod p$

Pegando emprestado o Teorema Binomial de Newton,

temos

$$(b+1)^p = \overbrace{(b+1)(b+1)(b+1)\cdots(b+1)}^{p\ (b+1)'s}$$

$$= b^p + p\cdot b^{p-1} + \binom{p}{2} b^{p-2} + \cdots + p\,b^1 + 1$$

$$= \sum_{i=0}^{p} \binom{p}{i} b^{p-i}$$

$$\binom{p}{i} = \frac{p!}{i!\,(p-i)!} \quad \text{(aceitemos)}$$

Afirmação: Se $0 < i < p$, então $\binom{p}{i} \equiv 0 \bmod p$

De fato, $\binom{p}{i} = \dfrac{p!}{i!\,(p-i)!}$

Como $i < p$, temos que $i! = 1\cdot 2\cdot 3\cdot \ldots \cdot i$ não possui p como fator ✭ PFP

Como $i > 0$, então $(p-i) < p$, logo $(p-i)!$ também não possui p como fator! ✭ PFP

Logo $\binom{p}{i} = \dfrac{p!}{i!\,(p-i)!}$ ← tem fator p ← não tem fator p ✭ PFP

$= p\cdot x$

para algum $x \in \mathbb{Z}$, ou seja $\binom{p}{i} \equiv 0 \bmod p$

✭ PFP: Como p é primo, se p não divide nenhum dos fatores de um produto, também não pode dividir o produto como um todo.

Portanto

$$(b+1)^p \equiv \sum_{i=0}^{p} \binom{p}{i} b^{p-i}$$

$$\equiv b^p + b^0 + \sum_{i=1}^{p-1} \binom{p}{i} b^{p-i}$$

$\sum_{i=1}^{p-1} \binom{p}{i} b^{p-i} \equiv 0$ (pela afirmação acima!)

$$\equiv b^p + 1 \mod p.$$

■

Teorema: Sejam $n, k \in \mathbb{N}$ e $a \in \mathbb{Z}$.
Se $a^k \equiv 1 \mod n$ ← n não precisa ser primo!
então:

$$\forall l, m \in \mathbb{N} \left(l \equiv m \mod k \Rightarrow a^l \equiv a^m \mod n \right)$$

Prova: Exercício.

Corolário: Sejam p primo e $a \in \mathbb{Z}$ com $\text{mdc}(a, p) = 1$.
Então $\forall l, m \in \mathbb{N} \left(l \equiv m \mod n-1 \Rightarrow a^l \equiv a^m \mod n \right)$

Exemplos:

1) Um evento astronômico ocorreu hoje e se repete a cada 1023 anos.
Se você foi congelado hoje e será descongelado daqui a 2^{1000} anos, quantos anos terá que esperar após o descongelamento para que o tal evento ocorra?

Solução: Queremos saber o menor x inteiro maior que 2^{1000} que seja múltiplo de 1023.
Para isso basta saber o resto da divisão de 2^{1000} por 1023

Sabemos $2^{10} = 1024 \equiv 1 \mod 1023$
Como $1000 \equiv 0 \mod 10$, temos $2^{1000} \equiv 2^0 \mod 1023$
$\equiv 1 \mod 1023$
Logo teri que esperar $1023 - 1 = 1022$ anos até o evento ...

2) Calcular a forma reduzida de $7^{75} \bmod 37$.

Solução: 37 é primo e $mdc(7,37) = 1$, logo (PTF2)

$$7^{36} \equiv 1 \bmod 37$$

Como $75 \equiv 3 \bmod 36$, temos

$$7^{75} \equiv 7^3 \bmod 37$$

⋮ exercício ☺

✓

Teste de Primalidade de Fermat

1ª forma, baseada no PTF1

Entradas: $n \in \mathbb{N}$, $b \in \mathbb{Z}$ $(n \geq 2)$

Saída: a conclusão de que "n é composto" ou "teste inconclusivo"

Passo 1: testar se $b^n \equiv b \bmod n$.
Se sim, teste inconclusivo.
Se não, n é composto.

Teorema (Terminação do Teste de Fermat v1): ——

Prova: O algoritmo executa apenas 1 passo. ▨

Teorema (Corretude do Teste de Fermat v1): ——

Prova: O algoritmo "afirma" que n é composto no caso em que $b^n \not\equiv b \bmod n$.
Pela contrapositiva do PTF1 $\left(b^n \not\equiv b \bmod n \Rightarrow n \text{ não é primo}\right)$ isso está correto. ▨

Teorema (PTF2, escrita alternativa). Sejam $p \in \mathbb{N}$ e $a \in \mathbb{Z}$ com $a \not\equiv 0 \bmod p$.

Se p é primo, então $a^{p-1} \equiv 1 \bmod p$.

Teste de primalidade de Fermat v2

Entradas: $n \in \mathbb{N}$, $b \in \mathbb{Z}$

Saída: teste inconclusivo, ou a conclusão que n é composto

Passo 1: Se $b \equiv 0 \bmod n$ ou $b^{n-1} \equiv 1 \bmod n$, teste inconclusivo
senão: n é composto.

Teorema (Terminação do Teste de Fermat v2): —

Prova: O algoritmo executa apenas 1 passo. ▨

Teorema (Corretude do Teste de Fermat v2): —

Prova: O algoritmo "afirma" que n é composto no caso em que $b \not\equiv 0 \bmod n$ e $b^{n-1} \not\equiv 1 \bmod n$

Pela contrapositiva do PTF2 ($b^{n-1} \not\equiv 1 \bmod n \Rightarrow n$ não é primo) isso está correto. ▨

Exemplos

$2^{341} \equiv 2 \bmod 341$ ⇝ teste inconclusivo!

mas

$3^{341} \not\equiv 3 \bmod 341$ ⇝ 341 é composto

Ideia razoável: Dado n, vamos procurar base b com $0 \leq b < n$, tal que $b^n \not\equiv b \bmod n$

(naturalmente podemos pular $b=0$ e $b=1$.

Se n for ímpar, também podemos pular $b=n-1$ também

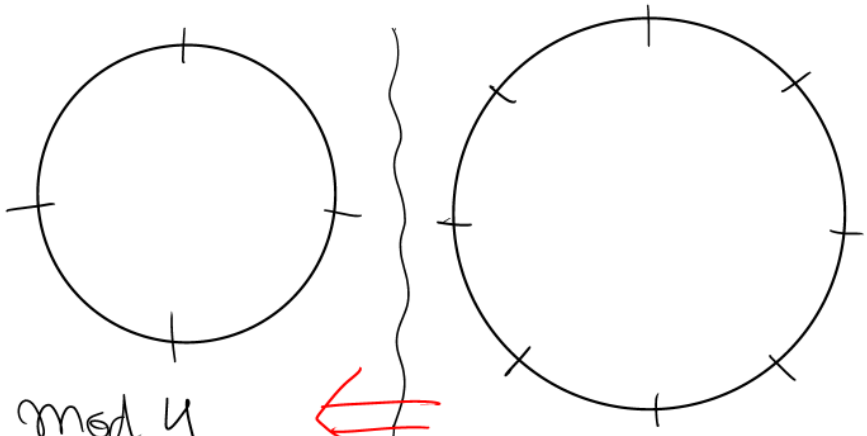
Infelizmente não resolve!

Def: Seja $n \geq 2$ natural e $b \in \mathbb{Z}$.
Se n é composto mas o teste de Fermat (V1) é inconclusivo para n com base b (ou seja, $b^n \equiv b \bmod n$), então chamamos n de pseudoprimo de fermat para a base b.

Ex: 341 é pseudoprimo de fermat para a base 2, mas não para a base 3.

Teorema (Carmichael & outros): Existem infinitos números naturais compostos que são pseudoprimos de fermat para todas as bases.

Tais números são chamados números de Carmichael



$a \equiv b \mod 4$ sse $4 \mid (a-b)$

$a \equiv b \mod 8$ sse $4 \mid 8 \mid (a-b)$

Teorema: Sejam $k, n \in \mathbb{Z}$, $k \neq 0 \neq n$.
Se $k \mid n$ então:

$$\forall a,b \in \mathbb{Z} \left(a \equiv b \mod n \Rightarrow a \equiv b \mod k \right)$$

Teorema (P.F.P. em linguagem de congruência):
Sejam $p \in \mathbb{N}$, $x, y \in \mathbb{Z}$.
Se p é primo, então:
(Se $xy \equiv 0 \mod p$ então $x \equiv 0 \mod p$ ou $y \equiv 0 \mod p$.)

Teorema (Lema § 2.6): Sejam $a, b, c \in \mathbb{Z}$, com $mdc(a,b) = 1$.

1) $ac \equiv 0 \mod b \Rightarrow c \equiv 0 \mod b$

2) $(c \equiv 0 \mod a \wedge c \equiv 0 \mod b) \Rightarrow c \equiv 0 \mod ab$

Teorema: Sejam $a, b, x, y \in \mathbb{Z}$

1) $x \equiv y \mod ab \Rightarrow (x \equiv y \mod a \wedge x \equiv y \mod b)$

2) $(mdc(a,b) = 1 \wedge x \equiv y \mod a \wedge x \equiv y \mod b)$
$\Rightarrow x \equiv y \mod ab$

Teorema (PFP, contrapositiva): Sejam $n \in \mathbb{N}$, $x, y \in \mathbb{Z}$
Se $xy \equiv 0 \bmod n$ & $x \not\equiv 0 \bmod n$ & $y \not\equiv 0 \bmod n$
então n não é primo

Teorema: Seja $n \in \mathbb{N}$.
Se existe $x \in \mathbb{Z}$ tal que $x^2 \equiv 1 \bmod n$
$\wedge\ x \not\equiv 1 \bmod n \ \wedge\ x \not\equiv -1 \bmod n$, então
n não é primo.

Prova: A hipótese é equivalente a
$\underbrace{x^2-1}_{(x-1)\cdot(x+1)} \equiv 0 \bmod n \wedge (x-1) \not\equiv 0 \bmod n \wedge (x+1) \not\equiv 0 \bmod n$
Logo, pela PFP vista acima, n não é primo.

Ex: $3^2 \equiv 1 \bmod 8$ mas $3 \not\equiv 1 \bmod 8$ & $3 \not\equiv -1 \bmod 8$, portanto 8 é composto.

"Moral da história": se encontrarmos $x \not\equiv \pm 1 \bmod n$ mas com $x^2 \equiv 1 \bmod n$, (★)
então n é composto!

A ideia do Teste de Miller-Rabin
Dado $n \in \mathbb{N}$ ímpar
$n-1 = 2^k \cdot q$, com $k > 0$ e q ímpar
↳ "parte ímpar"

Dado $b \in \mathbb{N}$ com $1 < b < n$,

temos $b^{n-1} = b^{(2^k \cdot q)} = \left(\left(\left(b^q \right)^2 \right)^2 \cdots \right)^2$ (k vezes)

O teste de Miller–Rabin consiste em verificar sucessivamente se algum dos

"chutes" $x = b^q$, $x = (b^q)^2$, $x = \left((b^q)^2 \right)^2$, etc, satisfaz as propriedades da "moral da história" (★) acima. Se algum desses chutes satisfizer, então n é composto!
Se o chute inicial $x = b^q$ for congruente a ± 1 mod n, ou algum outro chute for congruente a -1 mod n, o teste é inconclusivo.
Senão, ao fazermos o chute $x = b^{2^k \cdot q}$, concluímos

que n é composto (!), pois se n fosse primo, como $1 < b < n$, pelo PTF2 teríamos que ter $b^{2^k \cdot q} = b^{n-1} \equiv 1 \bmod n$

Algoritmo (Teste de Miller–Rabin)

Entradas: $n, b \in \mathbb{N}$ com n ímpar e $1 < b < n$

Saída: "teste inconclusivo" ou "n é composto"

Passo 1: calcular $k, q \in \mathbb{N}$ com $k > 0$ & q ímpar & $n - 1 = 2^k \cdot q$

Passo 2: chute $\leftarrow$ forma reduzida de $b^q \bmod n$.

Se chute $== 1$ ou chute $== n - 1$, retorne "teste inconclusivo"

Passo 3: para i de 1 até k:

chute $\leftarrow$ forma reduzida de chute$^2 \bmod n$

se chute $== 1$: retorne "n é composto"

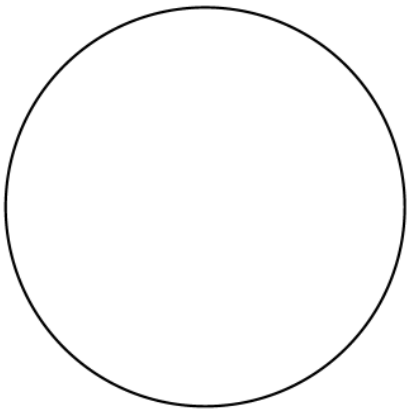
se chute $== n - 1$: retorne "teste inconclusivo"

Passo 4: retorne "n é composto"

—x—

Teorema (Rabin): Se $n \in \mathbb{N}$ é ímpar e composto, então para pelo menos 75% das bases b com $1 < b < n$, o teste de Miller–Rabin para n com base b comprova que n é composto!

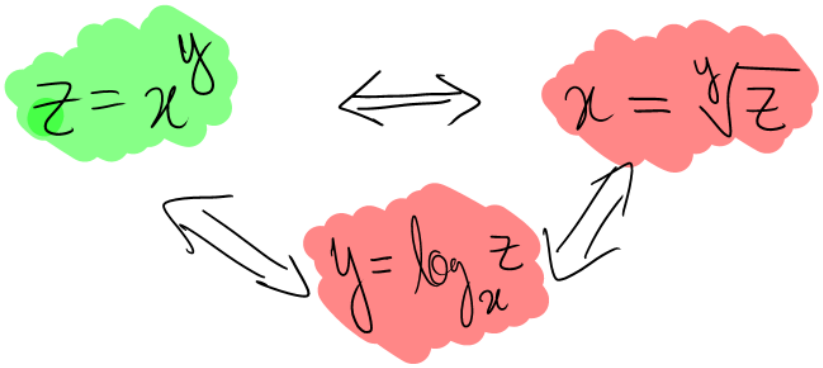
Dúvidas & comentários



cifra aditiva

" multiplicativa

Aritmética comum



Símbolos das mensagens

⇓ transformam, usando uma tabela

números (grandes)

'Hugo' ⇝ 177149134143

Investigando a ideia de encriptar e descriptar usando potências modulares

Temos algum módulo n e queremos expoentes e & d tais que $\forall b \in \mathbb{Z}$

e → encriptação

d → descriptação

$$(b^e)^d \equiv b^{e \cdot d} \equiv b \mod n \qquad (\star)$$

Tentativa 1: se n é primo, temos

PTF 1: $b^n \equiv b \mod n$

PTF 2: $\text{mdc}(b,n) = 1 \Longrightarrow b^{n-1} \equiv 1 \mod n$

→ poderia tentar $e \cdot d = n$, mas n é primo :^

→ para ter $b^{ed} \equiv b \mod n$, supondo $\text{mdc}(b,n) = 1$ basta que $e \cdot d \equiv 1 \mod n-1$

---

mas então é fácil obter d a partir de e (d é o inverso mult. de e mod $n-1$, pode ser obtido usando Euclides estendido :^)

Tentativa 2: se $n = p \cdot q$ com p, q primos distintos

Pergunta para a próxima aula: Como escolher e, d de forma a garantir $(\star)$?

LEMBRETE

Teorema: Sejam $a, b, x, y \in \mathbb{Z}$

1) $x \equiv y \bmod ab \Rightarrow (x \equiv y \bmod a \wedge x \equiv y \bmod b)$

2) $(mdc(a,b) = 1 \wedge x \equiv y \bmod a \wedge x \equiv y \bmod b)$
$\Rightarrow x \equiv y \bmod ab$

"trapdoor": informação extra "alçapão" (secreta) que facilita "desfazer"

Queremos formas de escolher n, e, d de forma que $\forall b \in \mathbb{N} \left((b^e)^d \equiv b \bmod n \right)$ (1) mas que seja impraticável encontrar d (2) a partir de n & e apenas.

Vimos: Com n primo é fácil escolher e, d que satisfaçam (1), mas não (2)

Hoje: $n = p \cdot q$ com $p \neq q$ primos.

Quero: escolher e, d tais que

(1): $\forall b \in \mathbb{N} \left(b^{ed} \equiv b \bmod pq \right)$

Basta que $\forall b \in \mathbb{N} \left(b^{ed} \equiv b \bmod p \right)$ (1a)

&

$\forall b \in \mathbb{N} \left(b^{ed} \equiv b \bmod q \right)$ (1b)

Para termos (1a):

Quero $\forall b \in \mathbb{N} \left(b^{ed} \equiv b \bmod p \right)$ (p: primo)

Tentativa: se $ed = p$ funcionaria, mas aí $e = p$ & $d = 1$, ou $e = 1$ & $d = p$, não queremos

↓ PTF 1

Note que, dado $b \in \mathbb{N}$, se $b \equiv 0 \mod p$, então $b^{ed} \equiv b \mod p$ para quaisquer e, d

E se $\underbrace{b \not\equiv 0 \mod p}_{mdc(b,p)=1}$? Então pelo PTF2 temos

$$b^{p-1} \equiv 1 \mod p$$

Então se $e \cdot d \equiv 1 \mod p-1$ teremos $b^{ed} \equiv b \mod p$!

Analogamente, para termos (1b), basta que

$$e \cdot d \equiv 1 \mod q-1$$

Para termos ambos 🟢 e 🔵, basta que

$$e \cdot d \equiv 1 \mod [(p-1)(q-1)]$$!

def: Dado $n = p \cdot q$ com $p \neq q$ primos, definimos

$$\phi(n) = (p-1) \cdot (q-1)$$

ϕ é chamada função totiente ou tociente de Euler.

O RSA

Geração de chaves

Cada usuário U que deseje receber mensagens em seguida deve:

1) Escolher primos distintos p, q de forma que

fatorar $n = p \cdot q$ seja impraticável

2) Calcular $\phi(n) = (p-1)(q-1)$ e escolher e, d inversos multiplicativos um do outro módulo $\phi(n)$ $\quad e \cdot d \equiv 1 \bmod \phi(n)$

3) Publicar a chave pública (n, e)

módulo público — expoente público

Na prática costumamos escolher e pequeno (o que implica em d grande), o que torna a encriptação mais fácil.

Além disso, no TF veremos que há mecanismos para facilitar/acelerar a descriptação.

4) Guardar em segredo a chave privada d

5) Jogar fora $p, q, \phi(n)$ (trabalho final: nem sempre ☺)

Envio de mensagens

Para enviar um texto T para um usuário U, deve-se:

1) Transformar T em um número K usando

algum método que U saiba desfazer!

2) Dada a chave pública (n, e) de U, quebrar o número K em blocos, cada um menor do que n (para que cada bloco seja sua própria forma reduzida mod n)

Por segurança, é importante que cada bloco seja grande (mas menor do que n)

3) Encriptar cada bloco b, fazendo a conta $b^e \bmod n$

4) Enviar os blocos encriptados para U, sem trocar sua ordem nem juntar! (o envio é feito em público)

Recebimento de mensagens

Ao receber uma sequência

$$C_0, C_1, C_2, \ldots, C_m$$

de blocos encriptados, o usuário U:

1) Descripta cada um usando sua chave privada, ou seja, para cada bloco C_i calcula

$$(C_i)^d \bmod n$$

2) Junta os resultados, obtendo K

3) Transforma para texto, obtendo T

Exemplo: $p = 101$ $q = 103$

$n = p \cdot q = 10403$

$\phi(n) = (p-1)(q-1) = 10200$

$e = 7$, $d = 8743$

Chave pública:
(10403, 7)

Chave privada
ou (10403, 8743)

Usando a tabela do TF, vamos enviar a mensagem "to pensando".

t	o	_	p	e	n	s	a	n	d	o
148	143	237	144	132	142	147	127	142	131	143

K = 148,143,237,144,132,142,147,127,142,131,143

blocos = 1481, 4323, 7144, 1321, 4214, 7127, 142, 1311, 43

encriptados = [3056, 4714, 2475, 8775, 10342, 6331, 3983, 3508, 7706]

— × —

Como transformar textos em números de forma que seja fácil desfazer?

Usar uma tabela de correspondência

símbolos ⟷ códigos (números)

não dá certo:

"ABC" ⇝ 123

"LC" ⇝ problema!!!

A	1	I	9
B	2	J	10
C	3	K	11
D	4	L	12
E	5	M	13
F	6	⋮	
G	7		
H	8		

outra ideia:

"ABC" ⇝ 11 12 13

ok!

"LC" ⇝ 2213

CAJA ⇝ 13 11 2011

13112 11

A	11	I	19
B	12	J	20
C	13	K	21
D	14	L	22
E	15	M	23
F	16	⋮	
G	17		
H	18		

Solução do Hugo: evitar o dígito 0 nos códigos

Como escolher p & q?

1) p, q têm que ser grandes
2) $|p-q|$ tem que ser grande (evitar fatoração por Fermat, livro seção 2.4)
3) $p-1, p+1, q-1, q+1$ não podem ter apenas fatores primos pequenos