Números Inteiros e Criptografia, PLE 2020

Lista de Exercícios 7

Submeta as soluções das questões marcadas com *
até 16 de outubro às 18:00 salvando um arquivo na sua pasta no Google Drive†

Justifique todas as questões.

Questão 1 (Reescrevendo expressões)**.** Em matemática, o uso de reticências (i.e., "$\cdots$" ou "$\ldots$") em expressões é bastante comum; por exemplo, a função $f : \mathbb{N} \to \mathbb{N}$ dada por

$$f(n) = \text{ a soma dos } n \text{ primeiros números naturais}$$

é comumente escrita da forma

$$f(n) = 1 + 2 + 3 + \cdots + (n-1). \qquad (\star)$$

Entretanto, o uso de reticências pode causar problemas de incerteza e ambiguidade, pois assume que o leitor será capaz de *deduzir* o conteúdo ocultado pelas reticências, o que pode não ser imediato. De fato, é bem questionável deduzir $f(0) = 0$ a partir da expressão $(\star)$.

Em geral, o uso de reticências esconde uma definição recursiva; *oficialmente* a função f acima é definida por

$$\begin{cases} f(0) = 0 \\ f(n) = f(n-1) + (n-1), \quad \text{para } n > 0. \end{cases}$$

Em cada item abaixo, reescreva a expressão que define $g : (\mathbb{N} \smallsetminus \{0\}) \to \mathbb{N}$ de forma recursiva, sem o uso de reticências (nem de *somatórios*, *produtórios* ou afins). (Lembrete: em cada item, o domínio da função g é o conjunto dos número naturais *exceto* o número 0).

* **a.** $g(n) = 1^2 + 2^2 + \cdots + n^2$

b. $g(n) = 1 + 3 + 5 + \cdots + (2n-1)$

c. $g(n) = 1^3 + 2^3 + \cdots + n^3$

* **d.** $g(n) = \dfrac{1}{1 \cdot 2} + \dfrac{1}{2 \cdot 3} + \cdots + \dfrac{1}{n(n+1)}$

* **e.** $g(n) = 2 \cdot 4 \cdot 6 \cdots (2n)$

†Link recebido por email em 1/9/2020 ou 17/9/2020. A pasta tem um nome similar a `Cripto - Submissões e Feedback - <seu nome>`; em caso de qualquer dúvida entre em contato com os professores.

*** f.** $g(n) = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdots p_n$, onde p_n é o n-ésimo primo (veja a Questão 7 para mais detalhes). (Você pode usar p_n na definição recursiva de g.)

g. $g(n) = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdots p$, onde p é o *maior* primo tal que $p \leq n$. (Logo $g(3) = 6 = g(4)$, por exemplo. Qual deve ser a definição do caso base $g(1)$ para que a definição recursiva *funcione bem*?)

*** h.** $g(n) = \sum_{i=1}^{n} f(i) = f(1) + f(2) + \cdots + f(n)$, onde $f : \mathbb{N} \to \mathbb{N}$ é uma função dada. (A sua resposta pode e deve usar f na definição recursiva de g).

Questão 2. Prove por indução que

*** a.** $1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$, para todo natural $n \geq 1$.

b. $1 + 3 + 5 + \cdots + (2n - 1) = n^2$, para todo natural $n \geq 1$.

c. $1^3 + 2^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{4}$, para todo natural $n \geq 1$.

*** d.** $n^2 < 2^n$, para todo natural $n \geq 5$.

e. $n^2 < n!$, para todo natural $n \geq 4$.

*** f.** $n^3 - n$ é divisível por 3, para todos naturais $n \geq 1$.

*** g.** $\mathrm{mdc}(F(n), F(n+1)) = 1$, para todo natural $n \geq 1$, onde $F(n)$ é o n-ésimo número de Fibonacci.

h. $3^n - 2$ é ímpar, para todo natural $n \geq 1$.

***Questão 3.** Encontre uma fórmula fechada (i.e., uma fórmula não-recursiva e que não use reticências, somatórios ou produtórios) para a seguinte expressão (em função de n) e depois prove por indução que a fórmula encontrada está correta para todo natural $n \geq 1$:

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \cdots + \frac{1}{n(n+1)}$$

Questão 4. Seja f uma função definida recursivamente:

$$\begin{cases} f(1) = 3 \\ f(k) = 7 \cdot f(k-1) & \text{se } k \geq 2 \end{cases}$$

Prove por indução que $f(n) = 3 \cdot 7^{n-1}$ para todos os naturais $n \geq 1$.

***Questão 5.** Seja g uma função definida recursivamente:

$$\begin{cases} g(1) = 1 \\ g(2) = 3 \\ g(k) = g(k-2) + 2 \cdot g(k-1) & \text{se } k \geq 3 \end{cases}$$

Prove por indução forte que $g(n)$ é ímpar para todos os naturais $n \geq 1$.

***Questão 6.** São dadas 3^n moedas de um real, uma das quais foi adulterada e pesa menos do que devia. Você tem uma balança de dois pratos mas não tem pesos; a única forma de pesagem permitida consiste em pôr algumas moedas em cada prato e verificar se a balança está equilibrada. Mostre, por indução, que n pesagens deste tipo são suficientes para achar a moeda adulterada.

Questão 7. Vamos denotar o n-ésimo primo por p_n. Assim $p_1 = 2$, $p_2 = 3$, $p_3 = 5$ por exemplo. O objetivo é achar uma cota superior para o n-ésimo primo em função de n.

a. Mostre que $p_{n+1} \leq p_1 \cdot p_2 \cdots p_n + 1$.

b. Use indução e o item anterior para mostrar que o n-ésimo número primo satisfaz a desigualdade $p_n \leq 2^{2^n}$.

Questão 8. Prove por indução que qualquer número natural $n \geq 24$ pode ser escrito como uma soma de 5's e 7's. Use o código abaixo para te ajudar.

```
def soma_com_cincos_e_setes(n):
  if n == 24:
      return [7,7,5,5]
  elif n == 25:
      return [5,5,5,5,5]
  elif n == 26:
      return [7,7,7,5]
  elif n == 27:
      return [7,5,5,5,5]
  elif n == 28:
      return [7,7,7,7]
  else:
      lista = soma_com_cincos_e_setes(n-5)
      lista.append(5)
      return lista
```

***Questão 9.** Prove por indução que qualquer número natural $n \geq 8$ pode ser escrito como uma soma de 3's e 5's.