

PRIMEIRA PROVA DE ANÁLISE DE ALGORITMOS
BCC, 1o. SEMESTRE DE 2006

Instruções:

1. A prova pode ser feita a lápis. Cuidado com a legibilidade.
2. Não é permitido o uso de folhas avulsas para rascunho.
3. Não é necessário apagar rascunhos no caderno de soluções.
4. Asserções imprecisas valem pouco. Justifique suas asserções (dentro do razoável!).

1. (i) [0.5 ponto] Prove que $\sum_{1 \leq k \leq n} \lfloor \log_2 k \rfloor = O(n \log n)$.
(ii) [0.5 ponto] Prove que $\sum_{1 \leq k \leq n} \lfloor \log_2 k \rfloor = \Omega(n \log n)$.
(iii) [2 pontos, bônus] Dê uma fórmula fechada para $\sum_{1 \leq k \leq n} \lfloor \log_2 k \rfloor$. Justifique.
2. Suponha que as funções f e g definidas sobre os inteiros positivos satisfazem as recorrências

$$f(1) = f(2) = 0 \quad \text{e} \quad f(n) = f(\lfloor n/3 \rfloor) + 1 \quad \text{para } n > 2. \quad (1)$$

e

$$g(1) = g(2) = 0 \quad \text{e} \quad g(n) = g(\lceil n/3 \rceil) + 1 \quad \text{para } n > 2. \quad (2)$$

- (i) [0.5 ponto] Faça uma tabela com os valores de $f(n)$ e $g(n)$ para valores de n pequenos (o suficiente para entender o comportamento destas funções).
- (ii) [0.5 ponto] Dê uma fórmula fechada para $f(n)$.
- (iii) [1 ponto] Prove que sua fórmula para $f(n)$ está correta.
- (iv) [1 ponto] Dê uma fórmula fechada para $g(n)$.
- (v) [1 ponto, bônus] Prove que sua fórmula para $g(n)$ está correta.
3. (i) [1 ponto] Sejam n e k inteiros positivos. Quanto é

$$\left\lfloor \frac{n}{k} \right\rfloor + \left\lfloor \frac{n+1}{k} \right\rfloor + \dots + \left\lfloor \frac{n+k-1}{k} \right\rfloor? \quad (3)$$

- (ii) [1 ponto] Suponha que $C(1) = C(2) = 0$ e, para $n > 2$, temos

$$C(n) = C\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + C\left(\left\lfloor \frac{n+1}{3} \right\rfloor\right) + C\left(\left\lfloor \frac{n+2}{3} \right\rfloor\right) + n. \quad (4)$$

Prove que $C(n) \geq n \lfloor \log_3 n \rfloor$.

- (iii) [1 ponto] Continuando com (ii), prove que $C(n) \leq n \lceil \log_3 \lceil n/2 \rceil \rceil$.
4. (i) [0.5 ponto] Consideremos o conjunto das permutações $v_1, v_2, \dots, v_n$ de $1 \dots n$ (tais permutações poderiam ser armazenadas no vetor $v[1 \dots n]$, com $v[i] = v_i$ para todo i). Suponha agora que $2 \leq i \leq n$. Quantas permutações $v_1, v_2, \dots, v_n$ de $1 \dots n$ são tais que v_i é o segundo maior elemento de $v_1, v_2, \dots, v_i$?

Considere agora o seguinte algoritmo para determinar o maior e segundo maior elemento do vetor $v[1..n]$ (supomos $n \geq 2$).

Algoritmo *Maior_2do_Maior*(v, n)

1. $maior \leftarrow \max\{v[1], v[2]\}$
2. $segmaior \leftarrow \min\{v[1], v[2]\}$
3. para i de 3 até n faça
4. se $v[i] > maior$
5. $segmaior \leftarrow maior; maior \leftarrow v[i];$
6. senão se $v[i] > segmaior$
7. $segmaior \leftarrow v[i]$
8. devolva $maior, segmaior$

Suponha que a entrada do algoritmo é uma permutação de $1..n$ escolhida uniformemente dentre todas as permutações de $1..n$.

- (ii) [0.5 ponto] Qual é o número esperado de atribuições efetuadas na linha 5 do algoritmo? Justifique.
- (iii) [1 ponto] Qual é o número esperado de comparações executadas na linha 6 do algoritmo? Justifique.
- (iv) [1 ponto] Qual é o número esperado de atribuições efetuadas na linha 7 do algoritmo? Justifique.

Nesta questão, você deve dar respostas exatas, isto é, não use a notação O nem a notação Ω .