MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2005 – 8 de abril

Primeira Prova

Nome do aluno: ______________________________ **Turma**:__________

Assinatura: ____

No. USP: __________________ **Professor**:__________________________

Instruções

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis.
3. A legibilidade também faz parte da nota!
4. A prova consta de 5 questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é necessário apagar rascunhos no caderno de questão mas especifique qual é a resposta e qual é o rascunho.
7. A prova é sem consulta.

Não escrever nesta parte da folha

Questão	Nota	Observação
1		
2		
3		
4		
5 (a)		
5 (b)		
Total		

Boa Sorte !

Questão 1 (Valor: 1.0)

Mostre que $\sum_{i=1}^{n} i^2 = \Theta(n^3)$.

Questão 2 (Valor: 1.0)

Qual é a ordem de crescimento da função $T(n)$ dada pela recorrência abaixo?

$$\begin{aligned} T(1) &= 1 \\ T(n) &= T(n-1) + n^2 \end{aligned}$$

(Em outras palavras, mostre a solução assintótica da recorrência acima.) Justifique a sua resposta.

Questão 3 (Valor: 2.5)

Considere o seguinte algoritmo recursivo para calcular o máximo de um vetor $v[p \mathinner{..} r]$.

Algoritmo Máximo (v, p, r)

1. se $p = r$
2.   devolva $v[p]$
3. senão
4. $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$
5. $m1 \leftarrow$Máximo (v, p, q)
6. $m2 \leftarrow$Máximo $(v, q+1, r)$
7.   se $m1 > m2$
8.   devolva $m1$
9.   senão
10.   devolva $m2$

Seja $C(n)$ o número de comparações executadas na linha 7 do algoritmo por uma chamada de Máximo(v, p, r), onde $n = r - p + 1$. Deduza do algoritmo uma recorrência que defina $C(n)$ e resolva-a, exibindo uma fórmula fechada para $C(n)$. (Ou seja, dê a solução exata da recorrência.)

Questão 4 (Valor: 2.5)

Considere o seguinte algoritmo que calcula o maior e o menor elemento de um vetor $v[1\,..\,n]$ com elementos distintos.

Algoritmo MaiorMenor (v, n)

1. *maior* $\leftarrow v[1]$
2. *menor* $\leftarrow v[1]$
3. para i de 2 até n faça
4.   se $v[i] >$ *maior*
5.   *maior* $\leftarrow v[i]$
6.   senão se $v[i] <$ *menor*
7.   *menor* $\leftarrow v[i]$
8. devolva *maior*, *menor*

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

Qual é o número esperado de comparações executadas na linha 6 do algoritmo? Qual é o número esperado de atribuições efetuadas na linha 7 do algoritmo? Justifique a sua resposta. (Apresente o valor exato. Nada de notação assintótica aqui.)

Questão 5 (Valor: $1.5 + 1.5$)

A *silhueta de um prédio* é uma tripla (l, h, r) de números positivos com $l < r$, onde h representa a altura do prédio, l representa a posição inicial da sua base e r a posição final.

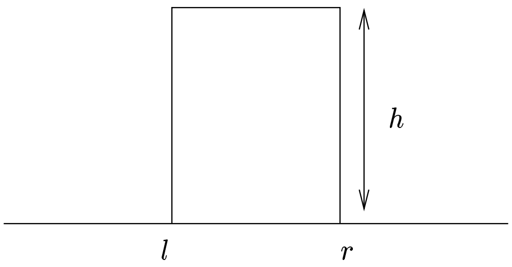

Figura 1: Silhueta (l, h, r) de um prédio.

Considere uma coleção $\mathcal{S} = \{(l_1, h_1, r_1), \ldots, (l_n, h_n, r_n)\}$ com a silhueta de n prédios. Para cada número positivo x, denote por $\mathcal{S}_x$ o conjunto $\{i : 1 \leq i \leq n \text{ e } \ l_i \leq x \leq r_i\}$. Denote ainda por $h(\mathcal{S}_x)$ o número $\max\{h_i : i \in \mathcal{S}_x\}$.

O *skyline* de $\mathcal{S}$ é uma seqüência $(x_0, t_1, x_1, \ldots, t_k, x_k)$ tal que

1. $x_0 = 0$ e $x_k = \max\{r_i : 1 \leq i \leq n\}$;

2. $x_{j-1} < x_j$ para $j = 1, \ldots, n$;

3. para $j = 1, \ldots, n$, $t_j = h(\mathcal{S}_x)$ para todo x tal que $x_{j-1} < x < x_j$;

4. $t_j \neq t_{j+1}$ para $j = 1, \ldots, n - 1$.

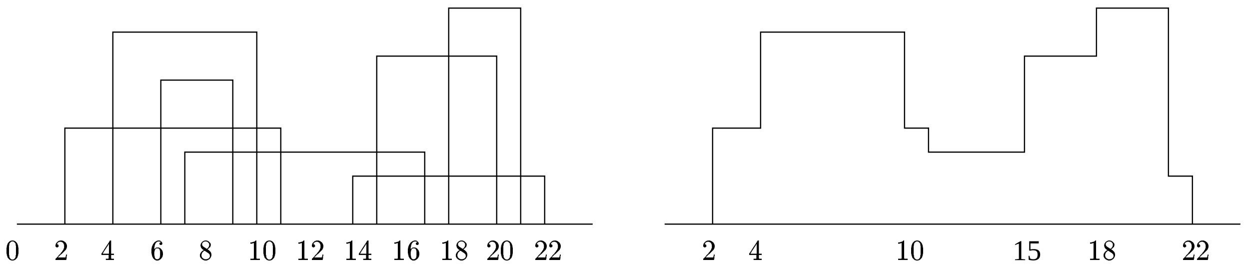

Figura 2: Coleção de silhuetas $\mathcal{S} = \{(15, 7, 20), (4, 8, 10), (18, 9, 21), (2, 4, 11), (7, 3, 17), (6, 6, 9), (14, 2, 22)\}$ e seu skyline $(0, 0, 2, 4, 4, 8, 10, 4, 11, 3, 15, 7, 18, 9, 21, 2, 22)$.

(a) **(Valor: 1.5)** Escreva um algoritmo que recebe o skyline de uma coleção $\mathcal{S}_1$ de silhuetas de prédios e o skyline de uma coleção $\mathcal{S}_2$ de silhuetas de prédios e devolve o skyline de $\mathcal{S}_1 \cup \mathcal{S}_2$. Seu algoritmo deve consumir tempo $O(n)$, onde $n = |\mathcal{S}_1 \cup \mathcal{S}_2|$. Explique por que seu algoritmo faz o que promete e por que consome o tempo pedido.

(b) **(Valor: 1.5)** Escreva um algoritmo que recebe um inteiro n e uma coleção $\mathcal{S}$ de n silhuetas de prédios e devolve o skyline de $\mathcal{S}$. Seu algoritmo deve consumir tempo $O(n \log n)$. Explique por que seu algoritmo faz o que promete e por que consome o tempo pedido.