MAC 6711 - Tópicos de Análise de Algoritmos

Departamento de Ciência da Computação
Primeiro semestre de 2021

Lista 1

1. Escreva um algoritmo que ordena uma lista de n itens dividindo-a em três sublistas de aproximadamente $n/3$ itens, ordenando cada sublista recursivamente e intercalando as três sublistas ordenadas. Analise seu algoritmo concluindo qual é o seu consumo de tempo.

2. Se $T(0) = T(1) = 1$, cada uma das seguintes recorrências define uma função T nos inteiros não-negativos.

   (a) $T(n) = 3T(\lfloor n/2 \rfloor) + n^2$;
   (b) $T(n) = 2T(n-2) + 1$;
   (c) $T(n) = T(n-1) + n^2$.

   Qual delas não pode ser limitada por uma função polinomial? Justifique a sua resposta.

3. Problemas 3-1 a 3-5 do CLRS.

4. Exercício 4.3-2 e problema 4-1 do CLRS.

5. Considere o seguinte algoritmo recursivo para calcular o máximo de um vetor $v[p\,..\,r]$.

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

   Seja $C(n)$ o número de comparações executadas na linha 6 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.)

6. Lembre-se do problema de determinar o número de inversões de um vetor. Naquele problema, dada uma sequência de números $a_1, \ldots, a_n$ todos distintos, definimos uma inversão como um par de índices (i, j) tal que $i < j$ e $a_i > a_j$.

   A motivação para este problema é que o número de inversões é uma boa medida de quão diferentes duas ordens são. Entretanto, pode-se achar que essa medida é muito sensível. Assim, vamos chamar de uma *inversão significativa* um par de índices (i, j) tal que $i < j$ e $a_i > 2a_j$.

   Escreva um algoritmo $O(n \lg n)$ que, dado uma sequência de números $a_1, \ldots, a_n$ todos distintos, determina o número de inversões significativas da sequência. Como sempre, mostre que seu algoritmo funciona e deduza o seu consumo de tempo.

7. Descreva um algoritmo que, dados inteiros n e k, juntamente com k listas ordenadas que em conjunto tenham n registros, produza uma única lista ordenada contendo todos os registros dessas listas (isto é, faça uma *intercalação*). Use de divisão e conquista. O seu algoritmo deve ter complexidade $O(n \lg k)$. Note que isto se transforma em $O(n \lg n)$ no caso de n listas de 1 elemento, e em $O(n)$ se só houver uma lista (de n elementos).

8. 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.

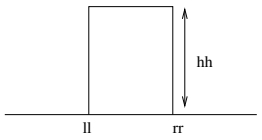

Figura 1: Silhueta (l, h, r) de um prédio.

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 sequência $(x_0, t_1, x_1, \ldots, t_k, x_k)$ 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 $j = 1, \ldots, n-1$.

(a) 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) 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 $O(n \lg n)$. Explique por que seu algoritmo faz o que promete e por que consome o tempo pedido.

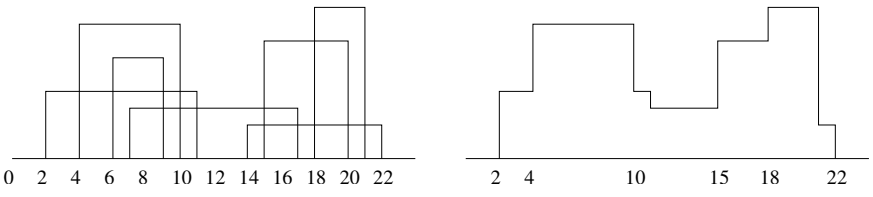

Figura 2: Coleção de silhuetas $\mathcal{S} = \{(15, 7, 20), (4, 8, 10), (18, 9, 21), (2, 4, 11), (7, 3, 17), (6, 6, 9), (14, 2, 22)\}$ e seu skyline $(0, 0, 2, 4, 4, 8, 10, 4, 11, 3, 15, 7, 18, 9, 21, 2, 22)$.

9. Descreva um algoritmo que conta o número de inversões de uma permutação de 1 a n em tempo $O(n \lg n)$ usando uma ABB balanceada (por exemplo, uma AVL ou uma rubro-negra). Pense em guardar alguma informação extra em cada nó da árvore que o ajude a calcular o número de inversões enquanto controi a árvore. Explique como manter essa informação extra e como dela obter o número de inversões, de modo que o tempo total do algoritmo seja de fato $O(n \lg n)$.