Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Mestrado e Doutorado em Ciência da Computação

Construção e Análise de Algoritmos

Lista de exercícios 1

1. Prove as seguintes afirmações sobre notação assintótica:

- $n^3/100 - 25n^2 - 100n + 7$ é $\Omega(n^2)$ e $\Theta(n^3)$
- $77n^3 - 13n^2 + 29n - 5$ é $O(n^4)$ e $\Omega(n^3)$
- $34n\ log_7 n^2 + 13n$ é $\Omega(n)$ e $O(n^2)$

2. Resolva as seguintes equações de recorrência segundo o método da árvore de recursão:

- $T(n) \ = \ 2 \cdot T(n/3) \ + \ n$
- $T(n) \ = \ 3 \cdot T(n/3) \ + \ n$
- $T(n) \ = \ 4 \cdot T(n/3) \ + \ n$
- $T(n) \ = \ 8 \cdot T(n/3) \ + \ n^2$
- $T(n) \ = \ 9 \cdot T(n/3) \ + \ n^2$
- $T(n) \ = \ 10 \cdot T(n/3) \ + \ n^2$
- $T(n) \ = \ T(0.99 \cdot n) \ + \ n$
- $T(n) \ = \ T(0.99 \cdot n) \ + \ 7$
- $T(n) \ = \ a \cdot T(n^{1/a}) \ + \ log_b(n)$, onde a e b são inteiros maiores que 1

3. O algoritmo do k-ésimo mínimo ainda seria $\Theta(n)$ se tomássemos grupos de 3 elementos, ao invés de 5? E se tomássemos grupos de 7 elementos? Justifique usando o método da árvore de recursão.

4. Uma pessoa sobe uma escada composta de n degraus, com passos que podem alcançar entre 1 e $k \leq n$ degraus. Escrever equações de recorrência que permitem determinar o número de modos distintos da pessoa subir a escada.

5. Elabore um algoritmo para resolver o seguinte problema: dado um vetor com n números inteiros positivos e um outro número inteiro positivo x, determine se existem ou não dois elementos cuja soma é igual a x (dica: divisão e conquista).

6. Elabore um algoritmo $O(n)$ de decomposição de um vetor S em trê subvetores. Esse algoritmo recebe como entrada, além do vetor S, um valor piv pertencente a S, e os índices p e r, $1 \leq p \leq r$. O algoritmo deve rearrumar os elementos em $S[p \ldots r]$ e retornar dois índices q_1 e q_2 satisfazendo as seguintes propriedades:

(a) se $p \leq k \leq q_1$, então $S[k] < piv$;

(b) se $q_1 < k \leq q_2$, então $S[k] = piv$;

(c) se $q_2 < k \leq r$, então $S[k] > piv$.

7. Sejam $X[1 \ldots n]$ e $Y[1 \ldots n]$ dois vetores ordenados. Escreva um algoritmo $\Theta(log\ n)$ para encontrar a mediana de todos os $2n$ elementos nos vetores X e Y. Prove esta complexidade.

8. Seja $X[1 \ldots n]$ um vetor de inteiros. Dados $i < j$ em $\{1, \ldots, n\}$, dizemos que (i, j) é uma inversão de X se $X[i] > X[j]$. Escreva um algoritmo $\Theta(n\ log\ n)$ que devolva o número de inversões em um vetor X.

9. Elabore um algoritmo $\Theta(n\ log\ n)$ que, dado um vetor S com $n > 0$ elementos, retorna um vetor V de tamanho n com a seguinte propriedade: $V[i]$ é o número de ocorrências de $S[i]$ em S. Prove esta complexidade.

10. Altere o algoritmo HeapSort para trabalhar com Heaps mínimos, ao invés de Heaps máximos. Argumente porque é melhor trabalhar com Heaps máximos ao invés de Heaps mínimos.

11.

12. Prove usando loops invariantes que o algoritmo HeapSort e seu algoritmo da questão anterior estão corretos (dica: para cada algoritmo, prove a correção do pré-processamento e depois a parte final).