MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2016

Lista 5

1. Desenhe a árvore de decisão para o SelectionSort aplicado a $A[1\,..\,3]$ com todos os elementos distintos.

2. **(CLRS 8.1-1)** Qual a menor profundidade (= menor nível) que uma folha pode ter em uma árvore de decisão que descreve um algoritmo de ordenação baseado em comparações?

3. Mostre que $\lg(n!) \geq (n/4)\lg n$ para $n \geq 4$ sem usar a fórmula de Stirling.

4. **(CLRS 8.1-3)** Mostre que não há algoritmo de ordenação baseado em comparações cujo consumo de tempo é linear para pelo menos metade das $n!$ permutações de 1 a n. O que acontece se trocarmos "metade" por uma fração de $1/n$? O que acontece se trocarmos "metade" por uma fração de $1/2^n$?

5. **(CLRS 8.2-1)** Simule a execução do CountingSort usando como entrada o vetor

$$A[1\,..\,11] = \langle 6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2 \rangle.$$

6. **(CLRS 8.2-2)** Mostre que o CountingSort é estável.

7. **(CLRS 8.2-3)** Suponha que o **para** da linha 7 do CountingSort é substituído por

$$\textbf{para } j \leftarrow 1 \textbf{ até } n \textbf{ faça}$$

   Mostre que o CountingSort ainda funciona. O algoritmo resultante continua estável?

8. **(CLRS 8.2-4)** Descreva um algoritmo que, dados n inteiros no intervalo de 1 a k, preprocesse sua entrada e então responda em $\mathrm{O}(1)$ qualquer consulta sobre quantos dos n inteiros dados caem em um intervalo $[a\,..\,b]$. O preprocessamento efetuado pelo seu algoritmo deve consumir tempo $\mathrm{O}(n+k)$.

9. **(CLRS 8.3-4)** Mostre como ordenar n inteiros no intervalo de 0 até $n^2 - 1$ em tempo $\mathrm{O}(n)$.

10. **(CLRS 8.4-1)** Simule a execução do BucketSort com o vetor

$$A[1\,..\,10] = \langle 0.79, 0.13, 0.16, 0.64, 0.39, 0.20, 0.89, 0.53, 0.71, 0.42 \rangle.$$

11. **(CLRS 8.4-2)** Qual é o consumo de tempo de pior caso para o BucketSort? Que simples ajuste do algoritmo melhora o seu pior caso para $\mathrm{O}(n \lg n)$ e mantem o seu consumo esperado de tempo linear.