

MO417 — Complexidade de Algoritmos I — 2s2019

Lista de Exercício 5

Além dos exercícios abaixo, recomendo que façam a maior quantidade possível de exercícios dos livros texto (CLRS e Manber) dos capítulos relacionados.

Questão 1. (FKM) Dado uma matriz de números. Ordene cada linha da matriz. Em seguida, ordene cada coluna da matriz. Mostre que as linhas da matriz continuam ordenadas.

Questão 2. (Manber) Em alguns casos, a entrada de um algoritmo de ordenação já está quase ordenada, o que significa que o número de elementos fora de ordem é pequeno. Descreva como os algoritmos de ordenação que você conhece se comportam com sequências quase ordenadas. Que algoritmo você usaria?

Questão 3. (Manber) A entrada é um *heap* de tamanho n (em que o maior elemento está no topo), dado como um vetor, e um número real x . Projete um algoritmo para determinar se o k -ésimo maior elemento no heap é menor ou igual a x . No pior caso, seu algoritmo deve executar em tempo $O(k)$, independente do tamanho do heap. Você pode usar espaço de tamanho $O(k)$. (Note que você não tem que encontrar o k -ésimo maior elemento; você só precisa determinar sua relação com x .)

Questão 4. (Manber) A entrada são d sequências de elementos tais que cada sequência já está ordenada e há um total de n elementos. Projete um algoritmo $O(n \log d)$ para juntar todas as sequências em uma única sequência ordenada.

Questão 5. (Manber) A entrada é um conjunto S com n número reais. Projete um algoritmo de tempo $O(n)$ para encontrar um número que *não* está no conjunto. Mostre que $\Omega(n)$ é um limitante inferior no número de passos para esse problema.

Questão 6. (Manber) Dado um vetor de inteiros $A[1..n]$, tal que, para todo i , $1 \leq i < n$, temos $|A[i] - A[i + 1]| \leq 1$. Seja $A[1] = x$ e $A[n] = y$, tais que $x < y$. Projete um algoritmo de busca eficiente para encontrar j tal que $A[j] = z$ para um valor z , $x \leq z \leq y$. Qual é o número de comparações com z que seu algoritmo faz.

Questão 7. (Manber) Mostre usando árvore de decisão que o algoritmo que você desenvolveu no exercício anterior é ótimo no pior caso (ou melhore seu algoritmo até que você possa provar que ele é ótimo).

Questão 8. (Dasgupta et al., adaptado) Uma árvore binária completa com n nós pode ser representada por um vetor B indexado por elementos $1, 2, \dots, n$. Nesta questão, queremos representar uma árvore d -ária completa (em que cada nó que não é folha tem um número constante d de filhos e todas as folhas estão no nível mais abaixo) usando um vetor D .

(a) Dado um índice j , qual o índice de D corresponde ao pai de j ? E quais índices correspondem aos filhos de j ? Demonstre isso.

(b) Escreva um algoritmo linear para, dado um vetor D com n elementos, criar um *maxheap* d -ário. Defina a propriedade de *heap* correspondente e demonstre a correção do seu algoritmo e sua complexidade.

Questão 9. (Horowitz et al.) Os k -ésimos quantis de um conjunto de elementos são os $k - 1$ elementos que dividem o conjunto ordenado em k conjuntos de tamanhos quase iguais (a diferença de tamanho entre quaisquer dois conjuntos é no máximo um). Dê um algoritmo de tempo $O(n \log k)$ para listar os k -ésimos quantis de um conjunto.

Questão 10. Um professor irá fazer um jantar de confraternização com todos os alunos de sua sala em um restaurante. Como o restaurante é muito caro, alguns convites serão sorteados e os outros alunos precisam pagar se quiserem ir. Um aluno decide pagar pelo jantar se e apenas se pelo menos dois outros, um mais velho e um mais novo também tiverem confirmado (assim ele não fica chateado de ser o mais novo ou o mais velho na confraternização). Suponha que todos os alunos sorteados vão ao jantar e que cada aluno que ainda não decidiu ir tem a mesma chance em um sorteio. Quantos convites o professor precisa distribuir (assintoticamente)? Descreva a estratégia do professor como um algoritmo aleatorizado e calcule o número esperado de convites até que todos decidam ir. A estratégia é sortear um convite enquanto houver alunos ainda indecisos.

Dicas:

1. Você pode presumir que todas as idades são distintas, $e_1 < e_2 < \dots < e_n$.
2. Responda: o que acontece imediatamente após o primeiro sorteio? o que acontece se o primeiro sorteado for e_1 e segundo sorteado for e_n ? o que acontece se a ordem do sorteio for e_n, e_{n-1}, \dots, e_1 ?
3. Defina uma variável aleatória x_i que indica se um aluno foi sorteado ao final do processo.
4. Considere primeiro a situação em que um aluno decide pagar se e somente se houver um aluno mais velho que ele que já confirmou presença.

Questão 11. (Manber) Projete um algoritmo de divisão e conquista para encontrar o menor e o maior elementos de um conjunto. O algoritmo deve usar no máximo $3n/2$ comparações (para $n = 2^k$). Você pode apontar a razão desse algoritmo requerer menos que $2n - 3$ comparações do algoritmo trivial?

Questão 12. (Manber) A entrada é um conjunto S contendo n números reais e um número real x .

- (a) Projete um algoritmo para determinar se há dois elementos de S cuja soma é exatamente x . O algoritmo deve executar em tempo $O(n \log n)$.
- (b) Suponha agora que o conjunto S é dado de forma ordenada. Projete um algoritmo para resolver esse problema em tempo $O(n)$.