Lista de exercícios de fixação[1] – Ordenação – IC-UNICAMP – 1º semestre de 2017

Visão geral de algoritmos de ordenação

Questão 1. (FKM) Dado uma matriz de números, retangular. 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? (Você é encorajado a projetar o seu próprio algoritmo)

Fila de prioridade e Heapsort

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, seus 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. (CLRS) Exercícios: 6.1-1, 6.1-2, 6.1-3, 6.1-4, 6.1-5, 6.1-6, 6.2-1, 6.2-2, 6.2-6, 6.3-2, 6.4-1, 6.4-3,

Ordenação por particionamento

Questão 5. (CLRS) Exercícios: 7.1-1, 7.1-2, 7.2-2, 7.2-3, 7.2-4 (compare com **Questão 2.**), 7.2-5, 7.2-6(*),

Questão 6. (CLRS) Problemas: 7-4

Questão 7. (Manber) Construa um exemplo para o qual *quicksort* realiza $\Omega(n^2)$ comparações quando o pivo é escolhi tomando a mediana do primeiro, do último e do elemento do meio de uma sequência.

Ordenação em tempo linear

Questão 8. (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 inferiores no número de passos para esse problema.

Questão 9. (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 encontra 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 10. (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 11. (CLRS) Exercícios: 8.1-1, 8.1-2, 8.2-1, 8.2-4, 8.3-1, 8.3-3, 8.4-1, 8.4-2,

[1] Esta lista deve ser feita logo após as aulas do conteúdo correspondente e serve para fixar o conteúdo, confirmar ou identificar as dúvidas. Anote suas dúvidas e procure atendimento! Os exercícios são referências ou transcrições de exercícios dos livros-textos (CLRS/Manber), ou foram gentilmente cedidos por outros professores, particularmente por Flávio Keidi Miyazawa (FKM), Cid Carvalho de Souza e Orlando Lee (CID/OL).

Estatísticas de ordem

Questão 12. (Manber) Projete um algoritmo de divisão e conquista para encontra 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 13. (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 $O(n)$.

Questão 14. (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 15. (CLRS) Exercícios: 9.1-1, 9.2-1, 9.2-2, 9.2-4, 9.3-1, 9.3-2, 9.3-5, 9.3-7, 9.3-9,

Questão 16. (CLRS) Problemas: 9-1