

Instituto de Computação – UNICAMP
Complexidade de Algoritmos I – Turma A
Exercícios: **Ordenação**

- Os exercícios devem ser submetidos como um arquivo em formato PDF (digitado ou manuscrito digitalizado), no prazo estipulado, na página <http://www.ic.unicamp.br/~lehilton/mo417a/submit/>.
- Só serão aceitas listas com todas questões respondidas, mas serão corrigidos **apenas** os itens sorteados em <https://www.randomresult.com/ticket.php?t=248324RN5BT>.

Questão 1. Suponha que queremos ordenar um conjunto de n registros armazenados em disco. Se n for relativamente pequeno, a melhor estratégia costuma ser carregar todos os dados na memória RAM, ordená-los e gravá-los de volta. Se n for muito grande, então estratégias diferentes precisam ser consideradas. Considere duas situações: quando os dados são comparados por uma rotina dada como caixa-preta; e quando se deve ordenar por identificadores únicos, que são números de 1 até n . Pesquise e disserte brevemente (200/400 palavras) sobre algoritmos de ordenação nessas situações. Escreva um texto coeso (i.e., escreva um texto contínuo evitando dar duas respostas independentes) e considere as operações necessárias para os algoritmos citados e as implicações da sua estratégia com relação ao tempo de latência e cópia em disco, quantidade de memória RAM disponível, etc.

Questão 2. (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 3. (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 4. Um professor irá fazer 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.