

Universidade Estadual de Campinas – Instituto de Computação

MO417A – Complexidade de Algoritmos – 1º Semestre de 2020

<https://www.ic.unicamp.br/~lehilton/mo417a>

Tarefa 3

- As repostas das questões devem ser manuscritas, digitalizadas e submetidas como um arquivo em formato PDF. Escaneie ou utilize um aplicativo para corrigir a perspectiva de foto. O PDF deve ser submetido em <https://www.ic.unicamp.br/~lehilton/mo417a/submit/> com a chave fornecida até a data lá anotada.
- Cada questão deve estar em páginas separadas (preferencialmente até uma, no máximo duas), seja conciso. Escreva com uma letra legível, de tamanho razoável e com bastante espaçamento entre as linhas de forma a permitir anotações. Faça um rascunho e depois passe a limpo.
- Só serão aceitas listas com todas questões respondidas, mas serão corrigidas **apenas** aquelas sorteadas na página <https://www.randomresult.com/ticket.php?t=1690937AB7E8>.

Questão 1. Dada 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) (6.29) 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 3. (CLRS) (5.2-4) Use variáveis aleatórias indicadoras para resolver o seguinte problema, conhecido como problema do hat-check. Cada um dos n clientes entrega um chapéu a um verificador de chapéu em um restaurante. A pessoa que confere o chapéu devolve os chapéus aos clientes em uma ordem aleatória. Qual é o número esperado de clientes que recebem seu próprio chapéu?

Questão 4. Um professor irá fazer um jantar de confraternização com os estudantes de sua sala. Como o restaurante é muito caro, apenas alguns estudantes sorteados receberão convite; os outros precisam pagar se quiserem ir. Um estudante decide pagar pelo jantar se, e apenas se, pelo menos outros dois estudantes, um mais velho e um mais novo, confirmarem antes (assim ele não fica chateado de ser o mais novo ou o mais velho na confraternização). Suponha que todos os estudantes sorteados decidam participar e que cada estudante que ainda não decidiu tem a mesma chance em um sorteio. A estratégia do professor para que todos participem é sortear um convite entre os indecisos até que todos decidam participar. Quantos convites o professor precisa distribuir? Você pode supor que todas as idades são distintas.

- (a) Descreva a estratégia do professor na forma de um algoritmo aleatorizado que recebe as idades dos estudantes, e_1, e_2, \dots, e_n e devolve o conjunto de estudantes que receberão convites.
- (b) Estime o número esperado de convites (assintoticamente) que serão sorteados.

Dicas: considere primeiro a situação em que um estudante decide pagar se, e apenas se, algum estudante mais velho tiver confirmado presença antes.

Questão 5. (Skiena) O Sr. B. C. Dull acredita ter desenvolvido uma nova estrutura de dados para filas de prioridade que suporta as operações de Inserção, Máximo e Extrair-Máximo, todas com tempo de execução $O(1)$ no pior caso. Prove que ele está enganado.

Questão 6. (Skiena) Suponha que o vetor $A[1..n]$ tenha apenas números de $\{1, \dots, n^2\}$, mas que no máximo $\log \log n$ desses números aparecem no vetor. Projete um algoritmo que ordene A em tempo substancialmente menor que $O(n \log n)$. Calcule a complexidade e demonstre a correção do algoritmo.

Questão 7. (CLRS) (9.2-4) Suponha que usamos RandomizedSelect para selecionar o menor elemento de um vetor $\langle 3; 2; 9; 0; 7; 5; 4; 8; 6; 1 \rangle$. Descreva uma sequência de partições que resulte em uma execução de pior caso de RandomizedSelect.

Questão 8. (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.