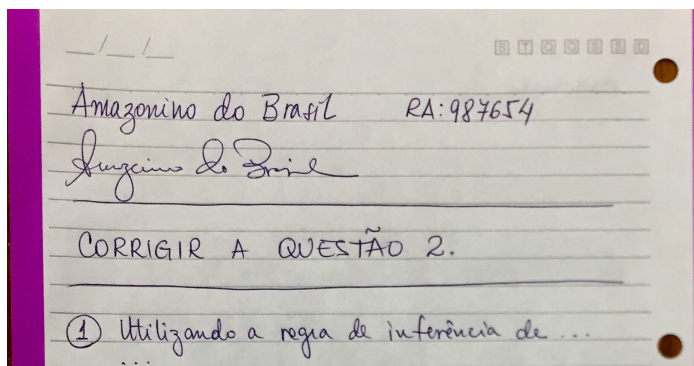


Lista Avaliativa 3

MC458 — Projeto e Análise de Algoritmos I
Prof. Pedro J. de Rezende
2º Semestre de 2020

Instruções

1. **Por se tratar de avaliação de conhecimentos adquiridos por cada aluno, a resolução desta Lista Avaliativa deve ser um trabalho individual sem consulta direta or indireta a outras pessoas.**
2. **QUALQUER TENTATIVA DE COLA OU FRAUDE ACARRETERÁ NOTA ZERO NESTA LISTA PARA TODOS OS IMPLICADOS, ALÉM DAS SANÇÕES PREVISTAS NO REGIMENTO GERAL DA UNICAMP (EM PARTICULAR, O ART. 227, INCISO VII, E OS ART. 228 A 231).**
3. Das três questões desta Lista, apenas duas serão corrigidas e valerão um total de 10 pontos.
 - Indique **exatamente UMA** das questões para ser corrigida pelo PED, a qual valerá nota de 0 a 5.
 - A segunda questão a ser corrigida será escolhida pelo PED, a qual também valerá nota entre 0 e 5. Se alguma questão estiver em branco, esta será a escolhida pelo PED.
4. O prazo **final** para submissão das resoluções se encerrará às 23hs do dia indicado no Google Classroom. Envios realizados após esse horário serão considerados atrasados. Se o atraso for de **até 2hs** após o encerramento do prazo regular de submissão, as resoluções submetidas serão corrigidas e receberão nota integral. Resoluções enviadas com **mais de 2hs de atraso**, mas tardias em **não mais do que 8hs** ainda serão corrigidas e receberão nota, mas com **50% de penalidade**. Submissões com atraso superior a 8hs automaticamente receberão nota zero.
5. **Importante:** note que a submissão não se completa apenas com o *upload* de um arquivo, mas **requer** o acionamento do botão “**Entregar**”. Como é essa ação que determina a hora do encaminhamento, sem ela a submissão não está efetivada e será desconsiderada.
6. **Justifique cuidadosamente todas as respostas.**
7. **Só serão aceitas** submissões de resoluções desta Lista Avaliativa na plataforma Google Classroom, e elas devem seguir **estritamente** o seguinte formato:
 - (a) As resoluções devem ser **manuscritas**, sem rasuras, escaneadas, formando **um único documento PDF cujo nome deve ser LEA#NNNNNN.pdf**, onde # é o número da presente LEA e NNNNNN é o seu R.A.
 - (b) No topo da primeira página das suas resoluções, coloque seu nome e RA de forma bem legível e, em seguida, a sua assinatura conforme esta consta em seu RG ou CNH. Veja modelo abaixo:



- (c) É **sua** responsabilidade **garantir** que o arquivo escaneado seja **claramente legível**. Para isso, recomenda-se o uso de um aplicativo para celular (**Android** ou **iOS**) como **Adobe Scan** (ou **CamScanner** ou **Office Lens** ou similar) para escanear as páginas manuscritas e, em seguida, fazer os devidos ajustes de contraste. Esses Apps facilitam a inclusão de múltiplas páginas em um único PDF. Todas as páginas devem estar na posição “retrato”.
- (d) Submissões constituídas meramente de arquivos de fotos (**jpg**, **png**, etc.), serão desconsideradas e receberão nota zero.

Se C é um conjunto finito de números inteiros **distintos**, chamamos de *elemento mediano* de C a um elemento $c \in C$ tal que $|\#(C_{<}) - \#(C_{>})| \leq 1$ onde $C_{<} = \{x \in C | x < c\}$, $C_{>} = \{x \in C | x > c\}$ e $\#(C)$ representa a cardinalidade de C .

1. Sejam X e Y dois vetores ordenados com n números inteiros distintos cada um, tais que $X \cap Y = \emptyset$. Projete um algoritmo cuja complexidade de pior caso num modelo baseado em comparações seja $o(n)$ para encontrar um elemento mediano de $X \cup Y$.

É imprescindível que você prove a **corretude** e faça uma **análise da complexidade** de seu algoritmo.

2. Sejam X e Y dois vetores ordenados com n números inteiros distintos cada um, tais que $X \cap Y = \emptyset$. Encontre e **prove** a melhor cota inferior que você puder num modelo baseado em comparações para o problema de determinar um elemento mediano de $C = X \cup Y$.

Obs: Você pode assumir que a saída de qualquer algoritmo que resolve esse problema tem a forma (X, i) ou (Y, i) indicando que um elemento mediano de C está na posição i do vetor X ou Y , respectivamente.

3. Considere o seguinte problema.

Entrada: dados k vetores V_1, \dots, V_k contendo valores inteiros no intervalo $[1, n]$. Para $i \in \{1, \dots, k\}$, o vetor V_i tem tamanho n_i . Sabe-se que $k \in O(n)$ e $\sum_{i=1}^k n_i \in O(n)$.

Saída: os k vetores da entrada ordenados.

Note: a saída são os k vetores ordenados e não único vetor ordenado com os $\sum_{i=1}^k n_i$ valores!

Precisa-se de um algoritmo, descrito **em linguagem natural** – mas de modo claro e objetivo! – que resolva o problema proposto acima em tempo $O(n)$ e que use espaço total da ordem de $O(n)$. Como nenhum modelo computacional foi especificado, pode-se escolher livremente.

- Xitoró quer simplesmente aplicar HeapSort em cada um dos k vetores V_1, \dots, V_k .
- Chorãozinho quer simplesmente aplicar CountingSort em cada um dos k vetores V_1, \dots, V_k .

- (a) Justifique porque a solução de Xitoró não atende aos requisitos da questão.
 - (b) Justifique porque a solução de Chorãozinho não atende aos requisitos da questão.
 - (c) Elabore uma **solução completa** que atenda aos requisitos da questão e justifique cuidadosamente sua corretude e complexidades.
-