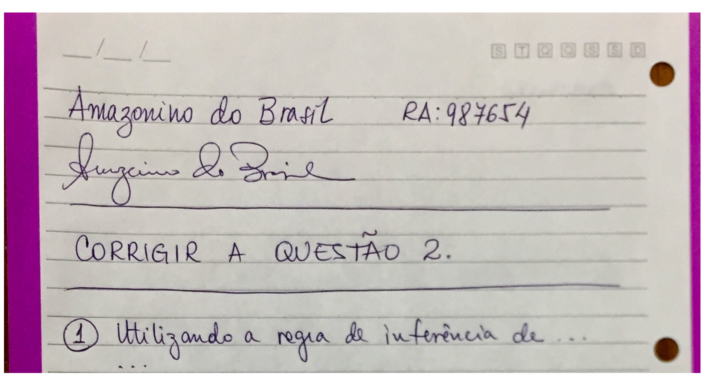
Lista Avaliativa 2

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 acarretará 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 <u>todas</u> 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.

Nas questões abaixo, considere as figuras apresentadas meramente como ilustrativas, com o objetivo de elucidar os enunciados das questões. **Elas não fazem são parte da tarefa a ser submetida.**

1. Nesta questão, assuma que n é uma potência de 2. Seja M uma matriz $n \times n$ de números reais tal que cada uma de suas linhas é uma sequência crescente, $M(i, j) < M(i, j + 1)$ para $1 \leq i \leq n$ e $1 \leq j \leq n-1$, assim como cada uma de suas colunas é uma sequência crescente, $M(i, j) < M(i+1, j)$ para $1 \leq i \leq n - 1$ e $1 \leq j \leq n$.
   Dado um número x e a matriz M, para se determinar se existem i e j, $1 \leq i, j \leq n$, tais que $x = M(i, j)$, podemos construir um algoritmo por divisão e conquista que, dependendo do resultado da comparação de x com o valor de $M(n/2, n/2)$, retorna `verdadeiro` ou descarta uma parte substancial da matriz antes de prosseguir recursivamente.

   Faça uma **prova por indução *forte*** de que é possível resolver o problema acima, de tal modo que de sua prova advenha o algoritmo acima esboçado e escreva a relação de recorrência que descreve a complexidade de pior caso do algoritmo resultante.

   Você **precisa apresentar a prova por indução** que dá origem ao algoritmo, **mas não precisa apresentar o pseudo-código do algoritmo** propriamente.

$$\begin{bmatrix} 4 & 8 & 9 & 12 & 15 & 18 & 19 & 20 \\ 5 & 9 & 10 & 21 & 24 & 30 & 78 & 138 \\ 7 & 10 & 14 & 100 & 105 & 130 & 139 & 141 \\ 17 & 19 & 24 & 132 & 203 & 205 & 210 & 220 \\ 19 & 24 & 132 & 203 & 300 & 400 & 401 & 500 \\ 29 & 34 & 203 & 204 & 301 & 401 & 500 & 511 \\ 39 & 44 & 232 & 233 & 310 & 402 & 510 & 520 \\ 49 & 54 & 233 & 234 & 400 & 500 & 511 & 521 \end{bmatrix}$$

Figura 1: Exemplo de uma matrix M coforme o enunciado da Questão 1.

2. Seja T uma árvore binária enraizada não vazia.

   Projete um algoritmo por indução *forte* que rotula cada vértice v de T com o valor absoluto da diferença $|\text{DFMP}(v) - \text{DFMD}(v)|$, onde $\text{DFMP}(v)$ é a Distância (em número de arestas) de v até a sua Folha mais Próxima e $\text{DFMD}(v)$ é a Distância (em número de arestas) de v até a sua Folha mais Distante. Se T tem apenas sua raiz, esta deve ser rotulada com zero. Denote por $v.D$ o campo de v que deverá receber o valor do rótulo.

   Você **precisa apresentar a prova por indução** que dá origem ao algoritmo, **mas não precisa apresentar o pseudo-código do algoritmo** propriamente.

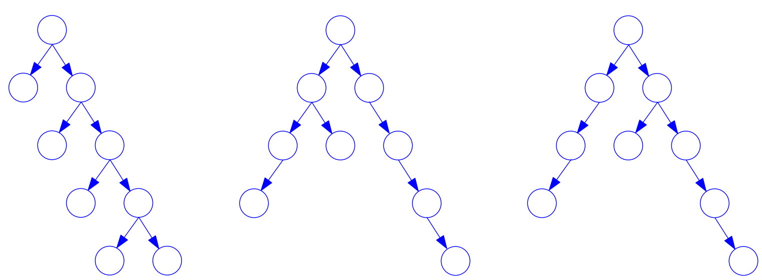


Figura 2: Exercite seu entendimento do enunciado da Questão 2 preenchendo os rótulos dos vértices destas árvores. **Cuidado**: não confunda esses rótulos com o conceito de *fator de balanceamento*.

3. Uma árvore ternária enraizada T cuja raiz contém um número **real** r é denominada uma *Árvore de Tusca* se T é vazia ou se as seguintes condições são verdadeiras:
   (a) Todo vértice da subárvore esquerda de T contém um **real** que é menor que r;
   (b) Todo vértice da subárvore central de T contém um **real** que é igual a r;
   (c) Todo vértice da subárvore direita de T contém um **real** que é maior que r;
   (d) As subárvores esquerda, central e direita de T são Árvores de Tusca.

   Projete um algoritmo por indução *forte* que, dada uma árvore ternária, verifica se ela é uma Árvore de Tusca.

   Você **precisa apresentar a prova por indução** que dá origem ao algoritmo, **mas não precisa apresentar o pseudo-código do algoritmo** propriamente.

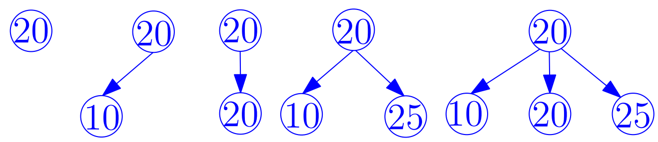


Figura 3: Cinco exemplos de Árvores de Tusca.

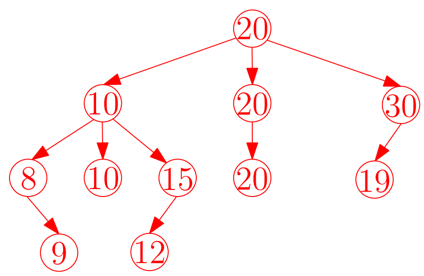


Figura 4: Há 1 vértice que impede que esta árvore ternária seja uma Árvore de Tusca.

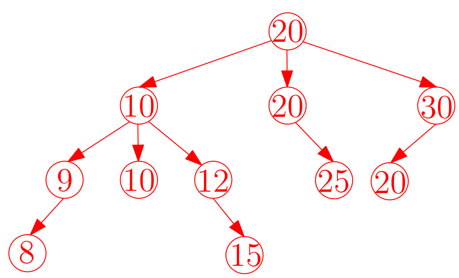


Figura 5: Há 2 vértices que impedem que esta árvore ternária seja uma Árvore de Tusca.

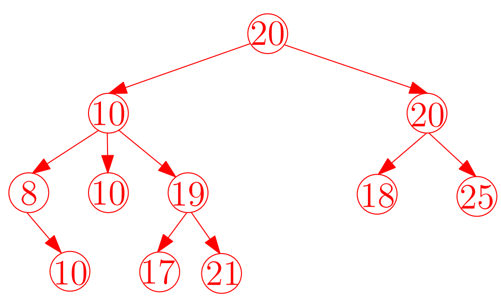


Figura 6: Há 4 vértices que impedem que esta árvore ternária seja uma Árvore de Tusca.