

Nome –  
Assinatura –

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
- 

- 1) **(2,0) Deseja-se ler uma sequência S de 0's e 1's e decidir qual dos dois símbolos ocorre com mais frequência em S (ou se houve empate). Exemplo: se S = 0101001 então a resposta é "0", pois ele ocorre 4 vezes (enquanto que "1" ocorre apenas 3 vezes). Descreva em palavras um algoritmo que, lendo os símbolos um por um, resolve este problema com o uso de uma pilha.**

R: Uma possível solução para este problema seria ler os símbolos comparando se o elemento no topo da pilha é igual ao símbolo corrente. Caso seja, a variável que indica a posição do topo é incrementada, caso contrário, decrementada. Ao final, se o valor desta variável for 0, significa que houve empate. Caso contrário, a pilha na posição topo indicará o ganhador. Vale lembrar que esta solução começa considerando o topo igual a zero.

- 2) **(2,0) Seja V um vetor contendo 5 elementos (números positivos) quaisquer.**  
**(a) (1,0) Descreva uma configuração de V que leva o algoritmo de ordenação por seleção a fazer o maior número possível de trocas entre elementos (pior caso).**

R: O algoritmo de ordenação por seleção é um algoritmo que possui a mesma complexidade em todos os casos:  $O(N^2)$ . Isto significa que, independente da configuração do vetor, o algoritmo sempre irá executar a mesma quantidade de trocas e comparações. Portanto, considerando um vetor com os elementos 6, 8, 9, 7, 5, qualquer configuração causaria o máximo possível de trocas.

- (b) (1,0) Descreva uma configuração de V que leva o algoritmo de ordenação pelo método da bolha a fazer o maior número possível de trocas entre elementos (pior caso).**

R: O Algoritmo de ordenação Bolha se baseia na ideia de que os elementos mais leves "sobem" e os elementos mais pesados "descem e vão para o fundo". Pode-se comparar a subida de um elemento como uma bolha, que tenta chegar a superfície. Neste sentido, um vetor cujos elementos estão organizados em ordem decrescente

apresenta o pior caso: Considerando os mesmos elementos anteriores, a configuração que com pior caso é: 9, 8, 7, 6, 5

3) **(2,0) Responda V ou F, justificando:**

**(a) (1,0) Toda árvore binária de busca cheia é uma árvore AVL.**

R: Uma árvore binária de busca cheia é uma árvore em que se um nó tem alguma sub-árvore vazia então ele está no último nível e é, portanto, uma folha. Partindo deste princípio podemos acusar a afirmação desta questão como sendo uma afirmação **verdadeira**. Isso acontece, pois uma árvore binária de busca cheia naturalmente estará balanceada, uma vez que a diferença entre as sub-árvores da esquerda e direita de um nó sempre será igual a zero.

**(b) (1,0) Se uma árvore B de ordem d e altura h possui d chaves em todas as suas páginas então o número de chaves desta árvore é o mínimo possível.**

R: Em uma Árvore B, o nó raiz pode ter entre 1 e 2d chaves, o que torna a afirmação **falsa**.

4) **(2,0) Deseja-se criar um sistema de atendimento na emergência de um hospital, da seguinte forma: toda pessoa que chega fica esperando sua vez, exceto nos seguintes casos: parturientes, cardíacos, pessoas com crise respiratória e pessoas em emergência ortopédica (estes quatro casos são prioritários). Explique qual é a estrutura de dados adequada para este sistema de atendimento, e como você a implementaria.**

R: Para a implementação deste sistema, o programador certamente deverá utilizar uma lista de prioridades que permita que pacientes sejam inseridos e removidos nesta lista dada a sua prioridade de atendimento. Há três formas de implementar esta lista de prioridade: a) uma lista não ordenada; b) uma lista ordenada) e c) um heap.

Utilizando uma lista não ordenada, a inserção na lista será simples, o paciente irá chegar e simplesmente ficar no lugar onde há vaga (mesmo com uma prioridade alta). O atendente do hospital, contudo, deverá buscar paciente por paciente, até encontrar o paciente que tem, a maior prioridade. Só então, o paciente poderá ser atendido. Portanto, em resumo: Inserção:  $O(1)$ , busca e remoção:  $O(N)$

Por outro lado, utilizando uma lista ordenada pela prioridade, a busca e remoção seriam  $O(1)$ , pois o paciente com maior prioridade já estará no início da fila. Contudo, a cada novo paciente que chegar no hospital, toda a lista terá de ser ordenada novamente  $O(N)$ .

Em contrapartida, a utilização de um heap permitirá a busca em  $O(1)$ , e inserção e remoção em  $O(\log N)$ . Isso acontece pois um heap garante que a seguinte propriedade seja atendida em todos os seus elementos

$$S_i \leq S_{i/2}$$

Onde,  $1 \leq i \leq n$

Portanto a estrutura de dados mais adequada para este sistema seria um heap de prioridades, onde, cada paciente que chegar ao hospital, seja inserido neste heap em  $O(\log N)$ . Quando houver médicos disponíveis para atendimento, a busca pelo paciente no heap será em  $O(1)$  e a remoção deste paciente da lista de espera será em  $O(\log N)$ .

5) **(2,0) Considere uma tabela de dispersão T com 7 endereços-base numerados de 0 a 6, utilizando a função de dispersão  $h(x) = x \bmod 7$ . Desejamos inserir em T as**

seguintes chaves (nesta ordem): 12, 23, 34, 45, 56, 67, 78, 89, 26, 49, 71. Suponha que T seja implementada utilizando encadeamento exterior. Desenhe a tabela resultante após a inserção das chaves acima.

