

Primeira Avaliação a Distância – Estrutura de Dados – 2º Semestre de 2018

1. Desenvolva um algoritmo **recursivo** que resolva o seguinte problema: Dada uma lista com n elementos ($n > 0$), na forma de um vetor $V[1..n]$, determinar o maior e o menor elementos desta lista. Calcule quantas chamadas recursivas o seu algoritmo efetua (incluindo a chamada externa). Pode haver elementos repetidos na lista.

2. Repita o exercício anterior, mas agora desenvolvendo um algoritmo **iterativo** para o problema. Calcule a complexidade do seu algoritmo contando o número de comparações efetuadas. (Uma comparação é uma verificação feita com um par de elementos, para decidir qual é maior e qual é menor, ou se há empate).

3. Escreva um algoritmo que leia um vetor com $n > 0$ elementos positivos e os classifique os em faixas de 3 em 3 elementos. A resposta deve ser da forma: existem x elementos na faixa 1 a 3; y elementos na faixa 4 a 6; z elementos na faixa 7 a 9 etc. Qual é a complexidade do seu algoritmo?

Exemplo: se o vetor de entrada contém os elementos 9, 5, 3, 2, 6, 17, 4, 10, 11, 12, a resposta será: existem **2** elementos na faixa 1 a 3; **3** elementos na faixa 4 a 6; **1** elemento na faixa 7 a 9; **3** elementos na faixa 10 a 12; **0** elementos na faixa 13 a 15; **1** elemento na faixa 16 a 18.

4. Escreva algoritmos de **busca, inserção e remoção** de um elemento em uma **lista simplesmente encadeada ordenada com nó cabeça**. Para cada algoritmo, determine sua complexidade.

5. Escreva um algoritmo que **inverte** uma **lista simplesmente encadeada com nó cabeça**, removendo os elementos repetidos.

Exemplo: se a lista de entrada contém os elementos 3, 5, 7, 3, 8, 3, 5, 9, 1, 2 (nesta ordem), a resposta do algoritmo será a lista contendo os elementos 2, 1, 9, 5, 3, 8, 7 (nesta ordem).

6. Escreva um algoritmo que leia uma sequência de votos onde cada voto tem apenas duas possibilidades (candidato A ou candidato B) e, ao término da leitura, determine qual é o candidato vencedor ou se houve empate. O seu algoritmo **não pode** fazer contagem de votos (por exemplo, usando dois contadores, um para cada candidato, e somando um ao contador correspondente de acordo com o voto lido na sequência). **Sugestão:** use uma **pilha**.

7. Os clientes chegam a um banco e vão formando uma fila de atendimento (fila “A”). Pessoas com prioridade formam uma fila à parte (fila “B”). A cada minuto chega um novo cliente. Suponha que há apenas um caixa atendendo, e que o atendimento é alternado (atende-se uma pessoa da fila A e depois da fila B, e assim por diante). Desenvolva um algoritmo que leia uma sequência formada por A’s e B’s (correspondendo a clientes que chegam para as filas A e B) e imprima o estado das filas, sabendo que cada atendimento de um cliente na fila A dura 2 minutos, e o tempo de atendimento de um cliente na fila B dura 3 minutos. Se uma fila fica vazia, pode-se atender dois ou mais clientes da outra fila em sequência.

Exemplo: se a sequência lida é AAABAABBA, o estado das filas será: (o cliente em negrito é o que está sendo atendido)

Tempo 0: **A** --

Tempo 1: **AA** --

Tempo 2: _ **AA** -- (note que o primeiro cliente saiu, e o segundo já começou a ser atendido)

Tempo 3: _ **AA** B

Tempo 4: __ **AA** B

Tempo 5: __ **AAA** B

Tempo 6: __ **AAA** BB

Tempo 7: __ **AAA** _ BB

Tempo 8: __ **AAAA** _ BB

Tempo 9: __ **AAA** _ **BB**

Tempo 12: ___ **AAA** ___ B

Tempo 14: ____ **AA** ___ **B**

Tempo 17: ____ **AA** ____

Tempo 19: _____ **A** _____

Tempo 21: _____ (FIM)

8. Escreva um exemplo de entrada com 10 elementos que leve o algoritmo de **ordenação por seleção** ao seu pior caso. Determine quantas comparações entre elementos o algoritmo efetua neste caso. Descreva, passo a passo, todas as trocas entre elementos efetuadas pelo algoritmo para ordenar a entrada.

9. Repita o exercício anterior, mas agora considerando o algoritmo de **ordenação pelo método da bolha**.

10. Desenvolva um algoritmo que ordene uma lista que só contenha dois tipos de valores, por exemplo 0 e 1. O seu algoritmo deve executar em tempo linear, isto é, $O(n)$ – onde n é o número de elementos a serem ordenados.

Exemplo: se a lista de entrada é 0, 1, 1, 0, 0, 1, 1, a saída será 0, 0, 0, 1, 1, 1, 1.