

#### Lista de Exercícios 4 - TAD e ordenação

- 1-) Utilizando o método bolha, quantas trocas são necessárias para que a sequência [66, 84, 4, 21, 71, 40] seja ordenada de modo crescente?
- 2-) Utilizando o método bolha, quantas trocas são necessárias para que a sequência [0 52 86 34 82 7 5] seja ordenada de modo decrescente?
- 3-) Utilizando o método de inserção, quantas trocas serão realizadas para ordenar em ordem crescente o vetor [7, 6, 5, 4, 3, 2, 1]?
- 4-) Utilizando o método de ordenação por seleção, qual será a ordem do vetor [72, 83, 17, 75, 31, 30] após o algoritmo realizar 3 trocas (ordenação crescente)?
- 5-) Aplicando o método quicksort no vetor [7, 11, 14, 6, 9, 4, 3, 12]. Considerando que o pivô selecionado é o primeiro elemento, qual será sua posição no vetor após a primeira iteração?
- 6-) O algoritmo de ordenação Merge sort utiliza qual das seguintes técnicas de programação?
  - (a) Divisão e Conquista
  - (b) Backtracking
  - (c) Algoritmo guloso
  - (d) Programação dinâmica
  - (e) Inteligência artificial
- 7-) Qual a principal desvantagem do algoritmo de ordenação merge sort em relação ao algoritmo de ordenação quick sort?
- 8-) Qual dos seguintes algoritmos de ordenação é a melhor opção para ordenar um vetor com 1 milhão de elementos?
  - (a) Quick sort
  - (b) Selection sort
  - (c) Insertion sort
  - (d) Ordenação direta
  - (e) Bubble sort
- 9-) Com base nas lógicas apresentadas anteriormente, tente criar a sua implementação do bubble, insertion, selection sort, quicksort e mergesort.
- 10-) Faça uma tabela relacionando as complexidades  $O(n)$  para os piores e melhores casos dos algoritmos de ordenação: Bubble sort, Insertion sort, Merge sort, Selection sort, Quick sort e merge sort.
- 11-) Considerando a estrutura de dados a seguir, assinale as operações que podem ser utilizadas como parte da definição do TAD triângulo.

```
typedef struct {  
  
    int lado1;  
  
    int lado2;  
  
    int lado3;
```

```
} triangulo;
```

Obs: Considere o nome, argumentos e tipo de retorno das funções para assinalar as alternativas

- a. triangulo \*criarTriangulo(int a, int b);
- b. float calculaArea(triangulo t1);
- c. int ehVazio(triangulo t1, triangulo t2);
- d. int soma(int l1, int l2, int l3);
- e. void calculaAltura(triangulo t);
- f. triangulo \*recebeTriangulo();
- g. int ehValido(triangulo \*t1);
- h. int ehEquilatero(triangulo \*a);

**12-) Considerando as variáveis definidas a seguir, assinale as alternativas corretas:**

```
typedef struct {
```

```
    int hora;
```

```
    int minuto;
```

```
    int segundo;
```

```
} horario;
```

```
horario *t1;
```

```
horario t2;
```

- a. scanf("%d", &t2.hora);
- b. scanf("%d", &t1.hora);
- c. printf("%d", t1);
- d. printf("%d", (\*t1).minuto);
- e. printf("%d", t1->hora);