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?

8 Trocas.

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?

9 Trocas.

3-) Utilizando o método de inserção, quantas trocas serão realizadas para ordenar em ordem crescente o vetor [7, 6, 5, 4, 3, 2, 1]?

21 Trocas.

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)?

[17, 30, 31, 75, 72, 83]

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?

3 posição.

6-) O algoritmo de ordenação Merge sort utiliza qual das seguintes técnicas de programação?

1. Divisão e Conquista
2. Backtracking
3. Algoritmo guloso
4. Programação dinâmica
5. Inteligência artificial

7-) Qual a principal desvantagem do algoritmo de ordenação merge sort em relação ao algoritmo de ordenação quick sort?

O uso do Merge Sort é desvantajoso levando em conta a sua necessidade de presença de um vetor auxiliar, que leva ao uso de memória adicional.

8-) Qual dos seguintes algoritmos de ordenação é a melhor opção para ordenar um vetor com 1 milhão de elementos?

1. Quick sort
2. Selection sort
3. Insertion sort
4. Ordenação direta
5. 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Código | Tempo | | |
|  | Melhor | Médio | Pior |
| Merge Sort | O(n log n) | O(n log n) | O(n log n) |
| Quick Sort | O(n log n) | O(n log n) | O() |
| Bubble sort | O(n) | O(x) | O(x) |
| Insertion Sort | O(n) | O(x) | O(x) |
| Selection Sort | O(x) | O(x) | O(x) |

11-) Considerando a estrutura de dados a seguir, assinale as operações que podem ser utilizadas como parte da definição do TAD triangulo.

typedef struct { int lado1;

int lado2; int lado3;

} triangulo;

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

1. triangulo \*criarTriangulo(int a, int b);
2. float calculaArea(triangulo t1);
3. int ehVazio(triangulo t1, triangulo t2);
4. int soma(int l1, int l2, int l3);
5. void calculaAltura(triangulo t);
6. triangulo \*recebeTriangulo();
7. int ehValido(triangulo \*t1);
8. 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;

1. scanf("%d", &t2.hora);
2. scanf("%d", &t1.hora);
3. printf("%d", t1);
4. printf("%d", (\*t1).minuto);
5. printf("%d", t1->hora);