Prova Prática

Para todas as modificações nos algoritmos de ordenação abaixo, considerar obrigatoriamente, os algoritmos visto em sala de aula.

Nome 1: _____

Nome 2: _____

1) Faça, usando linguagem C, uma função recursiva que calcule o valor da série S descrita a seguir para um valor n > 0 a ser fornecido como parâmetro para a mesma.

$$S = 2 + \frac{5}{2} + \frac{10}{3} + \ldots + \frac{1+n^2}{n}$$

2) Faça, usando linguagem C, a função recursiva que eleve qualquer número x a uma potência inteira n, sendo x e n pertencentes obrigatoriamente aos naturais.

Dado: $x^n = \begin{cases} 1 & \text{se } n = 0 \\ x * x^{n-1} & \text{se } n > 0 \end{cases}$

3) Faça, usando linguagem C, uma função recursiva que receba um número inteiro positivo N e retorne o fatorial quádruplo desse número. O fatorial quádruplo de um número N é dado por:

$$\frac{(2n)!}{n!}$$

4) Usando o algoritmo visto em aula, faça linguagem C o Bubble Sort invertido – Faça o algoritmo do bubblesort lterando-o para que funcione ao contrário (i.e., que resulte em números ordenados do maior para o menor).

5) Se duplicar o tamanho do vetor, o tempo que leva ao Bubble sort ordenar também duplica?

6) Refaça os algoritmos visto em aula, colocando estruturas para medir o tempo, a quantidade de comparações e a quantidade de trocas efetuadas.

7) Refaça o algoritmo Bubblesort excluindo do resultado final os elementos repetidos do vetor.

8) Comparar o número de trocas e comparações num vetor ordenado usando o algoritmo InsertSort e o algoritmo e o bublleSort

9) Usando o algoritmo visto em aula, faça linguagem C, faça o InsertionSort para ordenar em ordem decrescente.

Prova Prática

Para todas as modificações nos algoritmos de ordenação abaixo, considerar obrigatoriamente, os algoritmos visto em sala de aula.

Nome 1: _____

Nome 2: _____

10) Usando o algoritmo visto em aula, faça linguagem C o algoritmo do Selectsort lterando-o para que funcione ao contrário (i.e., que resulte em números ordenados do maior para o menor).

11) Se duplicar o tamanho do vetor, o tempo que leva ao Selectsort ordenar também duplica?

12) Refaça o Selectsort visto em aula, colocando estruturas para medir o tempo, a quantidade de comparações e a quantidade de trocas efetuadas. Compare a quantidade de trocas, comparações e tempo desse algoritmo com o BubbleSort e InsertSort

13) Utilizando o algoritmo shellsort visto em aula, implemente-o em C para que ordene em ordem crescente e obtenha o número de comparações e movimentações para o seguinte vetor: 45,56,12,43,95,19,8,67

14) Utilizando o algoritmo shellsort visto em aula, implemente-o em C para que ordene em ordem decrescente e obtenha o número de comparações e movimentações para o seguinte vetor: 8,12,19,43,45,56,67,95

15) Dado o seguinte vetor 45 56 12 43 09 19 08 17, mostre o passo a passo, no papel, a aplicação do algoritmo de ordenação ShellSort, aplicado para ordenação decrescente, onde h vale 4, 2 e 1.

16) Usando o algoritmo visto em aula, faça linguagem C o algoritmo de ordenação quicksort de modo que ele faça a ordenação em ordem decrescente.

17) Usando o algoritmo visto em aula, faça linguagem C o algoritmo de ordenação heapsort de modo que ele faça a ordenação em ordem crescente

18) Usando o algoritmo visto em aula, faça linguagem C o algoritmo de ordenação heapsort de modo que ele faça a ordenação em ordem decrescente

19) Dado o vetor 21 14 10 9 5, insira o elemento 12 no final desse vetor e demonstre no papel os passos necessários para a reconstrução do Heap máximo.

20) Usando o algoritmo visto em aula, faça linguagem C as modificações na função de ordenação MergeSort, de modo que ele faça a ordenação na ordem decrescente.