

Mateus Delai Takayama

1. Explique detalhadamente o funcionamento do algoritmo de ordenação: Insertion Sort

O Insertion Sort é um algoritmo de ordenação de números de um vetor. Seu funcionamento é dado pela análise se um número de posição i é maior que o de posição $i+1$, se sim eles trocam de local. Após isso, pega-se o $i+2$ e o analisa para ver se ele é menor que $i+1$, se sim, é analisado para ver se é menor que i , se sim, o $i+2$ ocupa a posição i e os outros são “empurrados” para a direita, se não for menor que o i , o $i+2$ ocupará a posição do $i+1$ e o empurrará para a direita, ordenando assim do menor para o maior.

Por exemplo: um vetor: {4, 3, 2}

- Primeiro é analisado o 4 e o 3, já que 4 é maior que 3 o 3 vai para a posição do 4 e empurra o 4 para direita.
- Após isso, resulta no vetor: {3,4,2}
- Agora analisa-se o número 2, ele é menor que o 4 e menor que o 3, então ele assume a posição do 3 e empurra todos os outros para a direita.
- Após isso, resulta no vetor: {2,3,4}, ou seja, ele se ordenou de maneira crescente.

2. Escreva o código do algoritmo de ordenação Insertion Sort na linguagem de programação C

CTRL + C e CTRL+V do código em C:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int vet[10];
    int MAX=10;
    int i, j, aux;

    for(int i=0; i<10;i++){
        printf("vet[%d]: ", i);
        scanf("%d" , &vet[i]);
    }

    printf("\n");
```

```

for(i = 1; i < MAX; i++){

    //auxiliar pega o valor da posição i
    aux = vet[i];

    //j assume o valor da posicao anterior a i
    // enquanto i-1 for maior que 0 e enquanto i-1 for maior que i (e
    decrescendo i-1)
    for(j = i-1; j >= 0 && aux < vet[j]; j--){

        //troca i por sua posição anterior
        vet[j+1] = vet[j];
    }
    //coloca o vet[i] no local correto
    vet[j+1] = aux;
}

for(int i=0; i<10;i++){
    printf("%d ", vet[i]);
}

return 0;
}

```

3. Compare os pontos positivos e negativos dos algoritmos Bubble Sort, Selection Sort e Insertion Sort.

Bubble Sort:

O Bubble Sort é um algoritmo de ordenação de números que faz comparações entre pares, se i for maior que $i+1$ eles trocam, depois compara $i+1$ com $i+2$ e assim por diante.

O ponto positivo é ser o código mais fácil de ser feito, pois tem uma lógica muito simples de ser feita.

Ponto negativo: faz muitas comparações e não ordena rapidamente de uma vez assim como o Insertion Sort.

Selection Sort:

O Selection Sort, assim como o Bubble Sort, é um algoritmo de ordenação de números, ele busca o menor número e o coloca na primeira posição, depois busca o segundo menor e o coloca na segunda posição e assim por diante.

Ele assim como o Bubble Sort é mais simples de ser feito e funciona muito bem, porém, assim como o Bubble realiza muitas comparações e muitas trocas de lugares.

Insertion Sort:

O Insertion Sort é um algoritmo de ordenação que é mais eficiente que os outros, porém, sua lógica é mais complexa e evita várias mudanças.

Assim como os outros realiza diversas mudanças de posições dos números, porém, menos que o Bubble e o Selection, sendo assim mais rápido que os outros métodos.