Insertion sort

- É um algoritmo de classificação simples que funciona de forma semelhante à maneira como você classifica as cartas de baralho em suas mãos. A matriz é virtualmente dividida em uma parte classificada e uma parte não classificada. Os valores da parte não classificada são escolhidos e colocados na posição correta na parte classificada.

Características do insertion sort

- Este algoritmo é um dos algoritmos mais simples com implementação simples
- Basicamente, a classificação de inserção é eficiente para valores de dados pequenos
- A classificação de inserção é de natureza adaptativa, ou seja, é apropriada para conjuntos de dados que já estão parcialmente classificados.

Análise de complexidade

- A complexidade de tempo da implementação recursiva do algoritmo de classificação de inserção é a mesma que a implementação iterativa, que é O(n^2).
- A classificação de inserção leva o máximo de tempo para classificar se os elementos forem classificados em ordem inversa. E leva um tempo mínimo (Ordem de n) quando os elementos já estão classificados
- A classificação de inserção é usada quando o número de elementos é pequeno. Também pode ser útil quando a matriz de entrada é quase classificada, apenas alguns elementos são extraviados na matriz grande completa.
- O algoritmo de Classificação de Inserção segue uma abordagem incremental.
- A classificação de inserção é um algoritmo de classificação in-loco
- A classificação de inserção é um algoritmo de classificação estável

Funcionamento do algoritmo de Classificação de Inserção:

Considere um exemplo: arr[ ]: {12, 11, 13, 5, 6}

12	**11**	**13**	**5**	**6**

Primeira passagem:

- *Inicialmente, os dois primeiros elementos da matriz são comparados na classificação de inserção.*

12	**11**	13	5	6

- *Aqui, 12 é maior que 11, portanto, eles não estão na ordem crescente e 12 não está em sua posição correta. Assim, troque 11 e 12.*
- *Então, por enquanto, 11 é armazenado em uma submatriz classificada.*

11	**12**	13	5	6

Segunda Passagem:

- *Agora, passe para os próximos dois elementos e compare-os*

11	**12**	**13**	5	6

- *Aqui, 13 é maior que 12, portanto, ambos os elementos parecem estar em ordem crescente, portanto, nenhuma troca ocorrerá. 12 também armazenados em um subarray classificado junto com 11*

Terceira passagem:

- *Agora, dois elementos estão presentes na submatriz classificada que são **11** e **12***
- *Avançando para os próximos dois elementos, que são 13 e 5*

11	12	**13**	**5**	6

- *Tanto o 5 quanto o 13 não estão presentes em seu lugar correto, então troque-os*

11	12	**5**	**13**	6

- *Após a troca, os elementos 12 e 5 não são classificados, portanto, trocam novamente*

11	**5**	**12**	13	6

- *Aqui, novamente 11 e 5 não são classificados, portanto, troque novamente*

5	**11**	12	13	6

- *Aqui, 5 está em sua posição correta*

Quarta passagem:

- *Agora, os elementos que estão presentes na submatriz classificada são **5, 11** e **12***

- *Passando para os próximos dois elementos 13 e 6*

5	11	12	**13**	**6**

- *Claramente, eles não são classificados, portanto, realizar a troca entre ambos*

5	11	12	**6**	**13**

- *Agora, 6 é menor que 12, portanto, troque novamente*

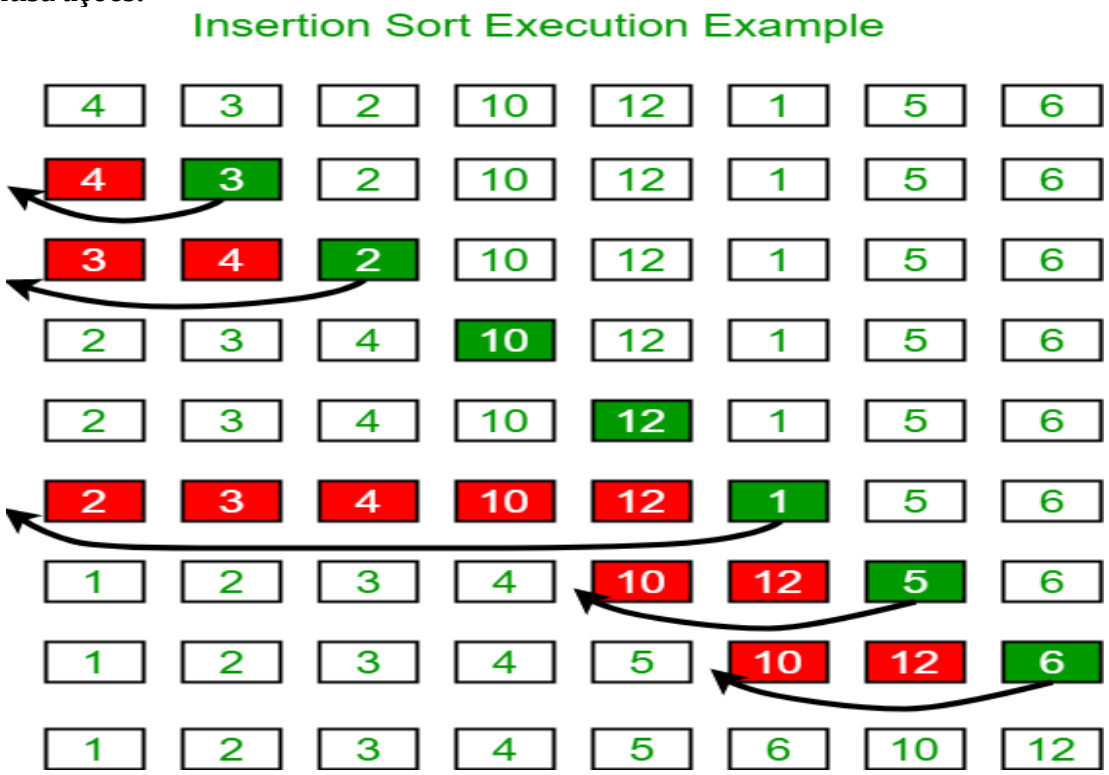
5	11	**6**	**12**	13

- *Aqui, também a troca faz 11 e 6 não classificados, portanto, trocar novamente*

5	**6**	**11**	12	13

- *Finalmente, a matriz é completamente classificada.*

Ilustrações:



Código

```
void insertionSort(int *lista, int tamanho){
    int i, j, aux;
    for ( i = 0; i < tamanho - 1; i++)
    {
        if (lista[i] > lista[i+1])
        {
            aux = lista[i+1];
            lista[i+1] = lista[i];
            lista[i] = aux;
            j = i-1;
            while (j>=0)
            {
                if (aux < lista[j])
                {
                    lista[j+1] = lista[j];
                    lista[j] = aux;
                }else
                {
                    break;
                }
                j=j-1;
            }
        }
    }
}
```

Bibliografia

Classificação de inserção - GeeksforGeeks

Método de ordenação Insertion Sort (implementação em linguagem C), Canal: Ponto Acadêmico(https://youtu.be/ECdLOLaIVx8)