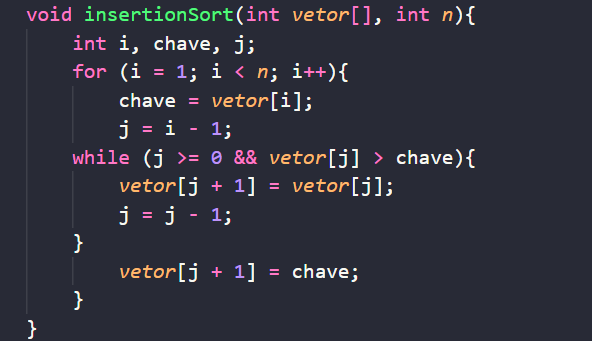
Insertion Sort.



Primeiro explicando como funciona o Insertion Sort, abaixo temos uma tabela para ficar mais fácil o entendimento.

Primeiro verificamos na tabela qual o primeiro número no caso o 3 e verificamos qual é o segundo número que no caso é o 1, com isso fazemos a seguinte comparação o 1 é menor que o 3 ? e sim o 1 é menor que o 3 portanto retiramos ele do nosso array e inserimos ele no local onde o 3 estava ou seja na posição 1 como mostra nas tabelas abaixo, contudo o 3 vai para a posição no qual o 1 se encontrava para começarmos a realizar a repetição de verificação no caso o 9 é menor que o 3 ? não o 9 é maior que o 3. Com isso continuamos verificando até encontrarmos um numero que é menor que o 3 para retirarmos ele do array e inserirmos na posição correta até chegarmos na lista totalmente ordenada.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01 | 02 | 03 | 04 | 05 | 06 |
| **3** | **1** | **9** | **5** | **2** | **8** |

**1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01 | 02 | 03 | 04 | 05 | 06 |
| **1** | **3** | **9** | **5** | **2** | **8** |

Agora explicando o Código (C) acima, no primeiro for verificamos qual o primeiro numero do indice fazendo j = i -1 com isso temos a primeira posição a ser verificada e também falamos que o j = o valor de indice -1 contudo o primeiro número a ser organizado passa a ser o j, para que no while realizamos as verificações caso o j for maior que o indice decrementamos j = j -1 fazendo com que ele saia do array e seja inserido na posição correta e se caso o j for maior que o indice fazemos com que a verificação continua acrescentando indice de j + 1 = chave no caso a chave é o vetor [índice].

Complexidade Temporal

Melhor caso -> O melhor caso ocorre quando o vetor está ordenado, pois, ao contrário do pior caso, a condição A[j] > elemento sempre será falsa. Logo, o código do laço interno nunca será executado. Assim, teremos apenas as n - 1 iterações do laço externo, portanto a complexidade no tempo é O(n)O(n).

Pior caso -> O pior caso do Insertion Sort ocorre quando os elementos do vetor estão em ordem decrescente, pois a condição A[j] > elemento sempre será verdadeira. Logo, o laço interno realizará a quantidade máxima de iterações. Nesse caso, o Insertion Sort terá complexidade no tempo de O(n2)O(n2).