

```
INSERTION-SORT(A)
1  for j ← 2 to length[A]
2      do key ← A[j]
3          ▷ Insert A[j] into the sorted
              sequence A[1 .. j - 1].
4      i ← j - 1
5      while i > 0 and A[i] > key
6          do A[i + 1] ← A[i]
7              i ← i - 1
8      A[i + 1] ← key
```

Entende-se que a letra A é a variável responsável por guarda o vetor que é recebido como argumento da função para ser ordenado.

Na **primeira linha** podemos notar um loop que percorre pela variável *j* com o valor 2 armazenado, por tanto, a cada interação o valor de *j* é alterado para o próximo número até o valor de A.

A **segunda linha** se trata de uma atribuição de um valor *A*[*j*] para a variável chamada *key*.

A **quarta linha** é uma função que define o valor da variável *i* que seria uma posição menor do que a posição atual.

A **quinta linha** possui a função de mover o valor chave em uma posição uma vez menor que a atual.

A **sexta linha** serve para substituir o valor atual para um valor posterior a ele.

A **sétima linha** apenas subtrai o valor de *i*.

Após está pequena análise para entender o que cada linha faz, podemos considerar que:

1ª linha: O tempo de execução que é definido pelo número de instruções (t) é apenas t , pois ele pode executar n vezes. Então, considerando o vetor $A = [3, 1, 4, 2, 5]$, a função iria executar 5 vezes.

2ª, 3ª, 4ª e 8ª linha: O número de instruções neste caso é apenas $t-1$, pois todos os próximos comandos executados dentro do loop, o que significa que poderá executar todas as vezes menos a última vez que não entrar no loop.

5ª linha: Temos um loop dentro de um loop, ou seja, ele será representado por um somatório.

$$\sum_{j=2}^t t_j$$

A soma mede quantas vezes elementos são deslocados ao longo da execução do algoritmo. Como cada deslocamento representa um passo computacional, essa soma se traduz diretamente no tempo de execução do algoritmo.

Exemplo com o vetor $A = [3, 1, 4, 2, 5]$:

- $j=2$ e $key=1$:
Se $A[1] = 3$ for maior que $key=1$ significa que temos 1 execução da linha 5.
- $j=3$ e $key=4$:
Se $A[2] = 3$ for maior que $key=4$ significa que temos 0 execução da linha 5.
- $j=4$ e $key=2$:
Se $A[3] = 4$ for maior que $key=2$ significa que temos 2 execuções da linha 5.
- $j=5$ e $key=5$:
Se $A[i] = 3$ for maior que $key=5$ significa que temos 0 execução da linha 5.

Se somarmos as execuções da linha 5, no caso, $1+0+2+0$, a linha 5 seria executada 3 vezes.

6ª e 7ª linha: são definidos pelo número de instruções:

$$\sum_{j=2}^t (t_j - 1)$$

Pois a linha 6 é executada t_j-1 vezes, porque deslocamos os elementos antes de inserir key . Já a linha 7 é ocorre **1 vez** por iteração, sempre ao final da inserção.

