Análise de Tempo de Execução - Insertion Sort

Algoritmo Insertion Sort:

```
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
```

Cálculo do Tempo de Execução (Pior Caso)

Contagem de tempo por linha no pior caso:

Linha 1: for j <- 2 to length[A]
- j = 2 -> 1 aritmética (t1)
- length[A] -> 1 acesso (t2)
- laço de repetição (n + 1) vezes com 3 operações básicas -> 3t(n + 1)

Total: 3t(n + 1)

Linha 2: key <- A[j]
- 1 aritmética + 1 acesso -> 2 operações por iteração -> 2tn

Linha 4: i <- j - 1
- 2 aritméticas -> 2tn

Linha 5: while i > 0 and A[i] > key
- 1 comparação lógica (i > 0) -> t
- 1 acesso + 1 lógica (A[i] > key) -> 2t Total: 3 operações × n -> 3tn

Linha 6: A[i + 1] <- A[i]
- 2 aritméticas + 2 acessos -> 4tn

Análise de Tempo de Execução - Insertion Sort

Linha 7: i <- i - 1

- 2 aritméticas -> 2tn

Linha 8: A[i + 1] <- key

- 2 aritméticas + 1 acesso -> 3tn

Organização e soma final:

= 3t(n + 1) + 2tn + 2tn + 3tn + 4tn + 2tn + 3tn

= 3tn + 3t + 16tn

= 19tn + 3t