

Análise de Tempo de Execução do Algoritmo INSERTION-SORT

Nome: Samuel Mello de Campos e Silva

Instituição: FATEC Antonio Russo - São Caetano do Sul

Disciplina: Estrutura de Dados

Professor: Carlos Henrique Verissimo Pereira

Data: 03 de junho de 2025

Pseudocódigo do INSERTION-SORT

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

Cada operação básica (atribuição, comparação, acesso a memória) consome 1t.

1, Loop externo (for $j \leftarrow 2$ to n), Executado $(n-1)$ vezes

2, $key \leftarrow A[j]$, 1t por iteração

4, $i \leftarrow j-1$, 1t por iteração

5, while $i > 0$ and $A[i] > key$, 2t por verificação (duas condições)

6, $A[i+1] \leftarrow A[i]$, 1t por iteração

7, $i \leftarrow i-1$, 1t por iteração

8, $A[i+1] \leftarrow key$, 1t por iteração

Análise do Pior Caso (Vetor em Ordem Decrescente)

O pior caso ocorre quando o vetor está invertido, forçando o deslocamento máximo de elementos no loop interno.

Cálculo do Loop Externo (Linhas 1-4, 8)

Número de iterações: $(n-1)$

Tempo por iteração:

Linha 2 ($\text{key} \leftarrow A[j]$): $1t$

Linha 4 ($i \leftarrow j-1$): $1t$

Linha 8 ($A[i+1] \leftarrow \text{key}$): $1t$

Total: $3t \times (n-1) = 3t(n-1)$

Cálculo do Loop Interno (Linhas 5-7)

Número máximo de iterações:

Para $j=2$: 1 iteração

Para $j=3$: 2 iterações

Para $j=n$: $(n-1)$ iterações

Total: $1 + 2 + \dots + (n-1) = n(n-1)/2$

Tempo por iteração:

Linha 5 (while): $2t$ (duas comparações)

Linha 6 ($A[i+1] \leftarrow A[i]$): $1t$

Linha 7 ($i \leftarrow i-1$): $1t$

Total por iteração: $4t$

Tempo total do loop interno:

$4t \times n(n-1)/2 = 2t(n^2 - n)$

Última Verificação do Loop While (Linha 5)

Ocorre uma verificação adicional para sair do loop.

Tempo: $2t \times (n-1)$

Tempo Total no Pior Caso

$$\begin{aligned} T(n) &= (\text{Loop externo}) + (\text{Loop interno}) + (\text{Verificação adicional}) \\ &= 3t(n-1) + 2t(n^2 - n) + 2t(n-1) \\ &= 2t n^2 + (3t - 2t + 2t) n - 3t - 2t \\ &= 2t n^2 + 3t n - 5t \end{aligned}$$

Simplificado: $T(n) = O(n^2)$

Análise do Melhor Caso (Vetor Já Ordenado)

O melhor caso ocorre quando o vetor já está ordenado, e o loop interno não executa (apenas verifica a condição uma vez).

Loop externo: $(n-1)$ iterações

Tempo por iteração:

Linha 2 ($\text{key} \leftarrow A[j]$): $1t$

Linha 4 ($i \leftarrow j-1$): $1t$

Linha 5 (while): $2t$ (apenas verificação)

Linha 8 ($A[i+1] \leftarrow \text{key}$): $1t$

Total: $5t \times (n-1) + t$ (última operação) $= 5t(n-1) + t$

Simplificado: $T(n) = O(n)$

Complexidade Assintótica

Pior caso, $2t n^2 + 3t n - 5t \rightarrow O(n^2)$

Melhor caso, $5t(n-1) + t \rightarrow O(n)$

Conclusão

O INSERTION-SORT é eficiente para pequenos conjuntos ou dados quase ordenados ($O(n)$), mas no pior caso (vetor invertido) seu tempo cresce quadraticamente ($O(n^2)$),

sendo menos eficiente que Quick-Sort para grandes entradas, devendo sua aplicação considerar o tamanho dos dados e sua organização prévia para otimização de desempenho.