Análise de Tempo do Algoritmo INSERTION-SORT

Nome: Matheus David Carbone Luna

Instituição: FATEC Antonio Russo - São Caetano do Sul Disciplina: Estrutura de Dados

Professor: Carlos Henrique Verissimo Pereira Data: 03 de junho de 2025

Texto, Carta

O conteúdo gerado por IA pode estar incorreto.

Considerando que cada operação básica (como atribuições, comparações, acessos à memória) consome um tempo t, a contagem de tempo é feita com base no número de vezes que cada uma dessas operações é executada.

# Loop Externo (Linhas 1, 2, 4 e 8)

O laço principal (for j ← 2 até n) é executado n-1 vezes. A cada iteração são realizadas as seguintes instruções:  
- Linha 2: key ← A[j] → 1 operação  
- Linha 4: i ← j-1 → 1 operação  
- Linha 8: A[i+1] ← key → 1 operação  
Total: 3t(n - 1)

# Loop Interno – Pior Caso (Linhas 5 a 7)

No pior cenário, com os elementos do vetor em ordem decrescente, o número de iterações do while cresce linearmente em cada passo do laço externo:  
- Iterações totais: 1 + 2 + 3 + ... + (n - 1) = n(n - 1)/2  
- Para cada iteração do while:  
 - Linha 5: verificação da condição → 2 operações (duas comparações)  
 - Linha 6: deslocamento de elementos → 1 operação  
 - Linha 7: decremento de i → 1 operação  
Total por iteração: 4t  
Tempo total: 4t × n(n - 1)/2 = 2t(n² - n)

# Verificações Adicionais do while

Mesmo ao sair do laço, há uma verificação final da condição em cada passo do loop externo:  
- Verificações extras: (n - 1) vezes  
- Custo: 2t(n - 1)

# Tempo Total no Pior Caso

Somando todas as partes:  
T(n) = 3t(n - 1) [loop externo]  
 + 2t(n² - n) [loop interno]  
 + 2t(n - 1) [verificações extras]  
  
T(n) = 2t n² + 3t n - 5t  
Complexidade assintótica: O(n²)

# Melhor Caso – Vetor já Ordenado

Nesse cenário, o while apenas verifica a condição e não entra no laço interno:  
- Operações por iteração do loop externo:  
 - Linha 2: 1 operação  
 - Linha 4: 1 operação  
 - Linha 5: 2 operações (verificação)  
 - Linha 8: 1 operação  
Total: 5t(n - 1)  
Verificação final: + t  
  
T(n) = 5t(n - 1) + t  
Complexidade assintótica: O(n)

# Conclusão

O algoritmo INSERTION-SORT apresenta boa performance em conjuntos pequenos ou já quase ordenados, com tempo linear no melhor caso. No entanto, para vetores desordenados em ordem reversa, o tempo de execução cresce quadraticamente, o que o torna menos eficiente do que algoritmos como o Quick-Sort em cenários com grandes volumes de dados. Sua utilização deve, portanto, levar em conta tanto o tamanho quanto a ordenação prévia dos elementos da entrada.