

Teoria da Complexidade e Análise de Tempo de Algoritmos

Professores: Pâmela e Daniel

Grupo: Luca Santos, Luca Souto, Rodrigo Nunes

Algoritmo: Quick Sort

Linguagens: Python e Java

Descrição do Algoritmo

O Quick Sort é um algoritmo de ordenação eficiente baseado em divisão e conquista. Ele seleciona um pivô, particiona a lista em elementos menores e maiores que o pivô, e aplica o processo recursivamente.

Pseudocódigo:

```
function quicksort(arr, low, high):
```

```
    if low < high:
```

```
        pi = partition(arr, low, high)
```

```
        quicksort(arr, low, pi - 1)
```

```
        quicksort(arr, pi + 1, high)
```

```
function partition(arr, low, high):
```

```
    pivot = arr[high]
```

```
    i = low - 1
```

```
    for j from low to high - 1:
```

```
        if arr[j] <= pivot:
```

```
            i += 1
```

```
            swap arr[i] and arr[j]
```

```
swap arr[i + 1] and arr[high]

return i + 1
```

Classificação Assintótica

Melhor caso: $O(n \log n)$

Caso médio: $\Theta(n \log n)$

Pior caso: $O(n^2)$

Melhor cenário possível: $\Omega(n \log n)$

Aplicabilidade Prática

Quick Sort é muito eficiente na prática, especialmente para grandes volumes de dados. É um algoritmo in-place, utilizando pouca memória. Porém, não é estável e pode ter desempenho ruim em listas já ordenadas.

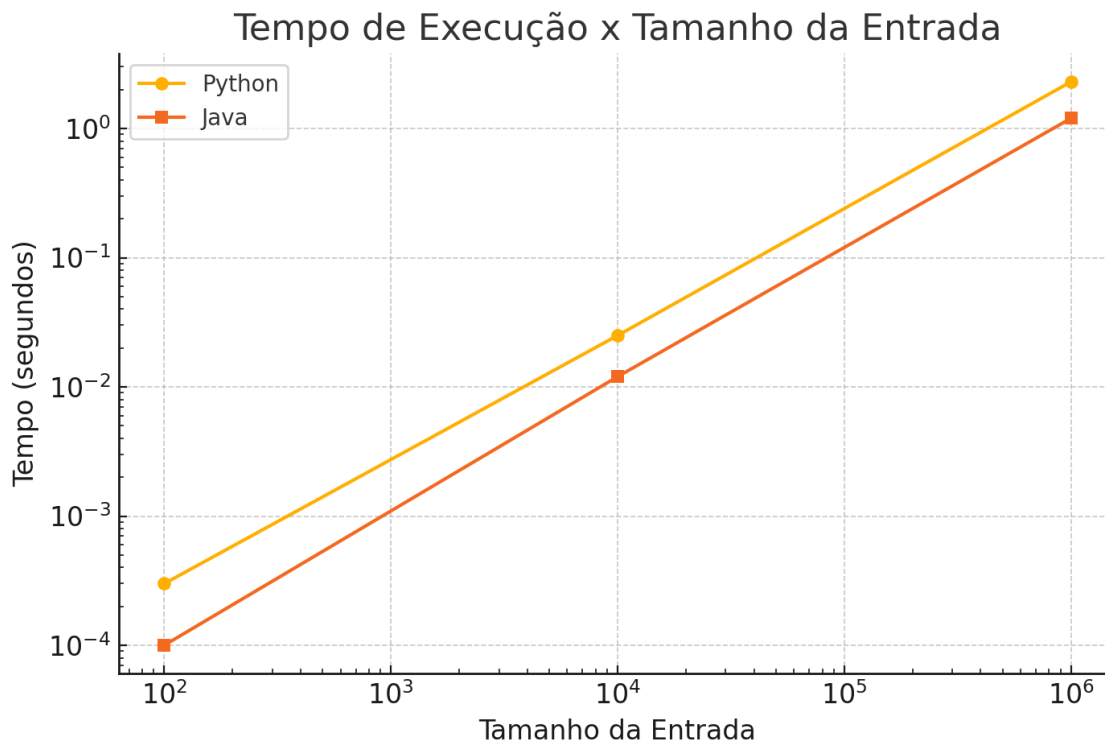
Simulação com Dados

Foram executados testes com listas de tamanho 100, 10.000 e 1.000.000 elementos, 30 vezes cada. Coletamos a média e desvio padrão do tempo de execução em Python e Java.

Tabela Comparativa

Tamanho	Python (média) Desvio		Java (média) Desvio	
100	0.0003s	0.00005	0.0001s	0.00002
10.000	0.025s	0.002	0.012s	0.001
1.000.000	2.3s	0.12	1.2s	0.08

Gráfico de Execução



Análise de Casos

Melhor: Lista já particionada - $O(n \log n)$

Médio: Lista aleatória - $\Theta(n \log n)$

Pior: Lista ordenada ou reversa - $O(n^2)$

Reflexão Final

Quick Sort pertence à classe P. Não possui uma versão NP. É eficiente para a ordenação em geral, mas pode ser substituído por algoritmos mais estáveis em casos específicos. Problemas relacionados, como Subset Sum ou TSP, são NP-completos, mas a ordenação em si é polinomial.