

Relatório - Performance de Ordenação

1. Objetivo

Comparar o desempenho de três algoritmos de ordenação em C: Bubble Sort, Insertion Sort e qsort.

A análise considera tempo de execução e número de comparações.

2. Metodologia

- Testamos vetores com 1000, 5000 e 10000 elementos
- Tipos de vetor:
 - * Aleatório
 - * Ordenado
 - * Decrescente
 - * Quase ordenado (~10% de elementos trocados)
- Cada teste repetido 5 vezes
- Calculamos mediana do tempo e média das comparações

3. Resultados

Bubble Sort

- Aleatório (1000): Tempo mediano = 0.78 ms, Comparações médias = 985014
- Ordenado (1000): Tempo mediano = 0.00 ms, Comparações médias = 999
- Decrescente (1000): Tempo mediano = 0.75 ms, Comparações médias = 999000
- Quase Ordenado (1000): Tempo mediano = 0.89 ms, Comparações médias = 896303

Bubble Sort

- Aleatório (5000): Tempo mediano = 16.62 ms, Comparações médias = 24655068
- Ordenado (5000): Tempo mediano = 0.01 ms, Comparações médias = 4999
- Decrescente (5000): Tempo mediano = 22.46 ms, Comparações médias = 24995000
- Quase Ordenado (5000): Tempo mediano = 21.49 ms, Comparações médias = 24207158

Bubble Sort

- Aleatório (10000): Tempo mediano = 94.39 ms, Comparações médias = 98320167
- Ordenado (10000): Tempo mediano = 0.01 ms, Comparações médias = 9999
- Decrescente (10000): Tempo mediano = 101.21 ms, Comparações médias = 99980001
- Quase Ordenado (10000): Tempo mediano = 74.64 ms, Comparações médias = 97464253

Insertion Sort

- Aleatório (1000): Tempo mediano = 0.39 ms, Comparações médias = 253687
- Ordenado (1000): Tempo mediano = 0.00 ms, Comparações médias = 999
- Decrescente (1000): Tempo mediano = 0.75 ms, Comparações médias = 499988
- Quase Ordenado (1000): Tempo mediano = 0.08 ms, Comparações médias = 60796

Insertion Sort

- Aleatório (5000): Tempo mediano = 7.76 ms, Comparações médias = 6318816
- Ordenado (5000): Tempo mediano = 0.01 ms, Comparações médias = 4999
- Decrescente (5000): Tempo mediano = 10.72 ms, Comparações médias = 12499974
- Quase Ordenado (5000): Tempo mediano = 1.58 ms, Comparações médias = 1433401

Insertion Sort

- Aleatório (10000): Tempo mediano = 28.31 ms, Comparações médias = 25175194
- Ordenado (10000): Tempo mediano = 0.01 ms, Comparações médias = 9999
- Decrescente (10000): Tempo mediano = 63.63 ms, Comparações médias = 50000028
- Quase Ordenado (10000): Tempo mediano = 6.53 ms, Comparações médias = 5783166

qsort

- Aleatório (1000): Tempo mediano = 0.04 ms, Comparações médias = 8710
- Ordenado (1000): Tempo mediano = 0.02 ms, Comparações médias = 4932
- Decrescente (1000): Tempo mediano = 0.04 ms, Comparações médias = 5261
- Quase Ordenado (1000): Tempo mediano = 0.04 ms, Comparações médias = 7907

qsort

- Aleatório (5000): Tempo mediano = 0.44 ms, Comparações médias = 55328
- Ordenado (5000): Tempo mediano = 0.17 ms, Comparações médias = 29804
- Decrescente (5000): Tempo mediano = 0.21 ms, Comparações médias = 32762
- Quase Ordenado (5000): Tempo mediano = 0.31 ms, Comparações médias = 51114

qsort

- Aleatório (10000): Tempo mediano = 1.02 ms, Comparações médias = 120533
- Ordenado (10000): Tempo mediano = 0.27 ms, Comparações médias = 64608
- Decrescente (10000): Tempo mediano = 0.41 ms, Comparações médias = 70540
- Quase Ordenado (10000): Tempo mediano = 0.61 ms, Comparações médias = 112248

4. Observações

- Bubble Sort é lento em vetores grandes e decrescentes
- Insertion Sort é rápido em vetores quase ordenados ou já ordenados
- qsort é rápido e consistente na maioria dos casos

5. Conclusão

O desempenho dos algoritmos varia conforme o tamanho e tipo do vetor:

- Vetores pequenos ou quase ordenados → Insertion Sort
- Vetores grandes ou desconhecidos → qsort
- Bubble Sort apenas para fins didáticos