**Relatório de Análise de Algoritmos de Ordenação – Murilo Pereira**

Este relatório apresenta uma análise de desempenho dos algoritmos de ordenação Bubble Sort, Insertion Sort e Quick Sort aplicados a três tipos de conjuntos de dados: aleatório, ordenado crescente e ordenado decrescente. Os tempos de execução são valores representados em nanosegundos (ns) para mostrar o comportamento comparativo entre os algoritmos.

**Resultados de Tempo de Execução**

|  |  |  |  |
| --- | --- | --- | --- |
| **Arquivo 100** | | | |
| Tipo de Conjunto de Dados | Bubble Sort | Insertion Sort | Quick Sort |
| Aleatório | 254.500 | 174.000 | 67801 |
| Crescente | 13.199 | 42.469 | 4.860 |
| Decrescente | 14.101 | 9.899 | 43.700 |

Este resultado demonstra que a eficiência do quicksort é muito melhor para um conjunto de dados aleatórios pelo grande nível de diferença entre os tempos, mas para outros conjuntos ele tem uma eficiência semelhante.

|  |  |  |  |
| --- | --- | --- | --- |
| arquivos 1000 | | | |
| Tipo de Conjunto de Dados | Bubble Sort | Insertion Sort | Quick Sort |
| Aleatório | 15.460.399 | 6.950.200 | 895.300 |
| Crescente | 1.009.300 | 1.292.000 | 1.908.501 |
| Decrescente | 3.175.200 | 459.299 | 1.255.999 |

Novamente o programa demonstra que no conjunto de dados com o arquivo de 1000 o quicksort foi mais eficiente com dados aleatórios, enquanto para outros conjuntos do tipo crescente e decrescente os resultados foram semelhantes.

|  |  |  |  |
| --- | --- | --- | --- |
| arquivos 10000 | | | |
| Tipo de Conjunto de Dados | Bubble Sort | Insertion Sort | Quick Sort |
| Aleatório | 340.507.600 | 68.997.400 | 2.347.000 |
| Crescente | 62.659.300 | 3.977.000 | 196.824.801 |
| Decrescente | 320.433.799 | 56.145.901 | 124.714.100 |

Como ilustrado, o Quick Sort, apresenta ser a melhor escolha para um conjunto de dados aleatórios, pois representa o menor tempo de execução entre eles, entretanto O Insertion Sort e o Bubble Sort podem ter resultados melhores em outros conjuntos de dados ordenados. Estes resultados exemplificam as diferenças gerais de eficiência esperadas entre os algoritmos em diferentes situações.