

Relatório Algoritmos de Ordenação

Arthur Barreto Godoi
Departamento de Informática
Universidade Federal do Paraná – UFPR
Curitiba, Brasil

Resumo—Este relatório visa comparar as diferenças entre algoritmos de ordenação e busca, em suas versões recursivas e iterativas.

I. INTRODUÇÃO

Neste relatório será apresentado as diferentes análises dos algoritmos de busca e ordenação, levando em consideração o tempo gasto na execução e o número de comparações realizadas. Os algoritmos analisados são respectivamente InsertionSort, SelectionSort, MergeSort, BuscaSequencial e BuscaBinária.

II. IMPLEMENTAÇÃO

Implementamos tanto as versões recursivas quanto iterativas dos algoritmos de ordenação, com exceção do mergeSort iterativo. Inicialmente, preenchemos um vetor de tamanho n com números aleatórios entre 0 e 100, utilizando a função `rand()`. Em seguida, ordenamos esse vetor utilizando todos os métodos de ordenação disponíveis.

Durante o processo de ordenação, medimos o tempo gasto utilizando a função `clock()`. Além disso, mantemos um ponteiro que é incrementado cada vez que o algoritmo realiza uma comparação entre elementos do vetor.

Todos os testes foram realizados no sistema operacional Linux Ubuntu com 8GB RAM.

III. TESTES DE ORDENAÇÃO

Para o primeiro teste foi utilizado um vetor de tamanho $n = 5000$

Algoritmo	Tempo	Comparações
Insertion Sort Iterativo	0.242177	5000
Insertion Sort Recursivo	0.233856	5000
Selection Sort Iterativo	0.434628	12497500
Selection Sort Recursivo	0.508139	12497500
Merge Sort Recursivo	0.013528	61808

Para o segundo teste foi utilizado um vetor de tamanho $n = 50000$

Algoritmo	Tempo	Comparações
Insertion Sort Iterativo	23.142742	50000
Insertion Sort Recursivo	25.267056	50000
Selection Sort Iterativo	45.293627	1249975000
Selection Sort Recursivo	47.621370	1249975000
Merge Sort Recursivo	0.121973	784464

Para o terceiro teste foi utilizado um vetor de tamanho $n = 150000$

Algoritmo	Tempo	Comparações
Insertion Sort Iterativo	207.677248	150000
Insertion Sort Recursivo	207.795405	150000
Selection Sort Iterativo	392.002849	11249925000
Selection Sort Recursivo	444.084474	11249925000
Merge Sort Recursivo	0.383831	2587856

Analisando as tabelas, observamos que não há diferenças significativas no número de comparações entre os algoritmos recursivos e iterativos. No entanto, notamos uma leve variação nos tempos de execução. Além disso, fica evidente que o algoritmo de ordenação selection sort se destaca por realizar uma quantidade substancialmente maior de comparações e, consequentemente, ser o mais lento dentre os comparados.

Por outro lado, o algoritmo de ordenação insertion sort realiza um número de comparações proporcional ao tamanho do vetor, mas, surpreendentemente, é mais rápido que o selection sort. Quanto ao mergesort, destaca-se como o mais eficiente em termos de tempo de execução, além de realizar uma quantidade razoável de comparações.

Em resumo, enquanto o selection sort se destaca por sua ineficiência em comparações e tempo de execução, o insertion sort surpreende ao ser mais rápido mesmo com uma quantidade linear de comparações. Por fim, o mergesort se sobressai como a opção mais rápida e eficiente dentre os algoritmos analisados.

IV. TESTES DE BUSCA

Para testar a eficiência dos algoritmos de busca foi feito dois testes no vetor $n = 150000$ já ordenados, procurando o valor 50 e 70 respectivamente, gerando as seguintes tabelas:

Valor = 50

Algoritmo	Tempo	Comparações	Índice
Busca Sequencial Iterativa	0.012146	75152	75151
Busca Sequencial Recursiva	0.023319	73368	76632
Busca Binária Iterativa	0.001207	13	76170
Busca Binária Recursiva	0.001386	13	76170

Valor = 70

Algoritmo	Tempo	Comparações	Índice
Busca Sequencial Iterativa	0.015919	105093	105092
Busca Sequencial Recursiva	0.016647	43352	106648
Busca Binária Iterativa	0.001500	11	105467
Busca Binária Recursiva	0.001616	11	105467

A busca sequencial, embora simples, pode ser notavelmente rápida, pois encontra o primeiro valor igual ao parâmetro de busca. No entanto, a busca binária apresenta uma leve vantagem em termos de velocidade. A diferença entre a busca sequencial iterativa e a recursiva reside na abordagem de busca: a iterativa começa da cabeça do vetor, enquanto a recursiva inicia da cauda.

Além disso, a busca binária reduz significativamente o número de comparações em comparação com a sequencial, especialmente em vetores ordenados. Essa característica torna a busca binária o algoritmo de escolha para pesquisas eficientes em vetores ordenados.

Ao analisar esses aspectos, fica claro que a busca binária se destaca como o método mais eficaz para a localização de elementos em vetores extensos e ordenados.

V. CONCLUSÃO

Com esse experimento foi possível observar na prática as diferenças nos custos dos algoritmos estudados em sala, além de presenciar como o custo do algoritmo influencia na sua eficiência.