Análise de Algoritmos de Ordenação e Busca

Gabriel Simon Batista Ribeiro
Departamento de Informática
Universidade Federal do Paraná – UFPR
Curitiba, Brasil
gabriel.simon@ufpr.br

***Resumo*—Análise sucinta de principais algoritmos de Busca e Ordenação, desenvolvimento de pensamento custo de operação.**

***Index Terms*—Comparação, pior custo, ordenação, busca, índice, recursão.**

I. Introdução

Analisar é uma tecnica extremamente necessaria para um cientista da computação, de modo que é necessário ela objetificando o estudo de seis algorítimos, dois deles de Busca de um índice dentro de um vetor: Busca Sequencial e Busca Binaria; quatro de Ordenação de vetores (InsertionSort, SelectionSort, MergeSort e QuickSort.

É fundamental saber outros conceitos e estabelecer metodos de analises, de forma cientifica, obtendo tirar devidas conclusões do algoritmo com menor custo (em pior caso). Quando conseguimos determinar o menor custo possível para resolver um devido problema, que no caso é a busca e ordenção, teremos uma maior eficiencia como um todo.

II. Metodo de Análise

Os métodos levados como base de análise são o custo de comparações e de tempo, na medida que o tamanho do vetor aumenta, de forma que obtemos resultados de comportamento dos algoritmos. Abstraindo em qual contexto que pode ser mais adequado a utilização dos mesmos, fermentado também o motivo de maior aplicação na ciência da computação.

Todos os presentes algoritmos passaram pelos seguintes atributos de avaliação, testados 15 vezes, com tamanhos de vetores de potência um até potencia 5, na base 10. Realizados em uma máquina de sistema operação Linux, distribuição popOS!, os tempos de execução de cada algoritmo foram velados em conta como análise, mas principalmente o número de comparação, o mais relevante durante a análise em questão.

III. Busca

Buscar um elemento dentro de uma lista de elementos, é uma ferramenta agudamente comum e fundamental no mundo da computação, mas encontramos empecilhos quando colocamos certos métodos para analisar os algoritmos de busca: o mais inicial e de certa forma bastante intuitivo, é a Busca Sequencial ou também Busca Ingenua, consiste em percorrer índice por índice de um vetor, parando quando encontra o elemento buscado, tem como problema justamente numero de comparações quando analisamos o pior caso.

A Busca Binaria é uma solução mais sofisticada, contendo uma lógica de fácil compreensão, para grandes vetores a diferença é expressivo o contraste de comparações quando colado ao lado com a Busca Ingenua, é indeficiente, como pode-se observar na Tabela I.

Tabela I
Relação Tamanho do Vetor/ Número de Comparações

	Vetor/Vaga por ano			
Tamanho Vetor	10^1	10^2	10^3	10^4
BuscaSequencial	3.4	6.72	9.97	13.36
BuscaBinaria	0.3	0.07	0.009	0.00085

IV. Ordenação

A ordenação se caracteriza como um pilar da computação, é um sistema de algoritmo largamente usado em diversos categorias de sistemas, contudo foram analisados os seguintes: InsertionSort, SelectionSort, MergeSort e QuickSort, talvez os mais comuns da área de ordenação de vetores. Conforme o gráfico abaixo, é observado o número de comparação, que no SelectionSort acaba por defasado, visto o alto número de comparação na medida que os vetores vão crescendo, isso muito em vista do seu pior caso é o pior entre os quatro Sort's.

É nítido a relevância e quantidade de comparação do MergeSort e do QuickSort, ambos são considerados os melhores no âmbito analisado, mas no caso do Merge em específico, surge um problema de ocupação de memória, pois o algoritmo se baseá na criação de sub-vetores, quando implementado em um grande array, certamente ocorrera problemas na memória.

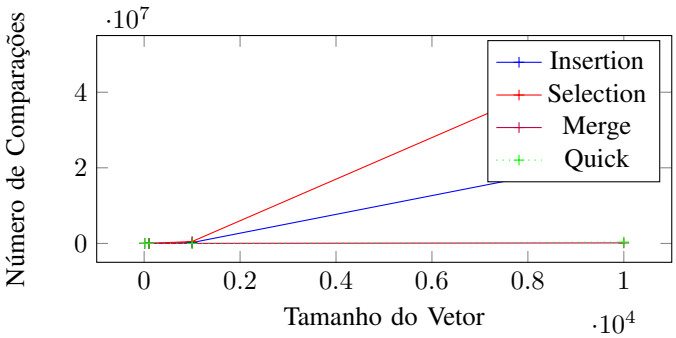

Figura 1. Gráfico com a relação Tamanho Vetor / Número de Comparação.

V. Conclusão

Contudo, obtivemos dados que constatam uma maior utilização da Busca Binário, faz de fato sentido, pois é matematicamente comprovado que não tem algoritmo de menor

custo, no quesito busca. Quando a ordenação, QuickSort é o mais popular até mesmo nas bibliotecas padrões de varais linguagens, também sendo o mais utilizado pela comunidade num geral, obviamente, podem ver contextos em que os outros algoritmos de solução se sobressaem, mas em um âmbito geral, estes dois citados anteriormente tem um custo muito mais agradável computacional.