

Relatório do Trabalho de Ordenação

Lucas Gabriel Batista Lopes

Universidade Federal do Paraná – UFPR

Algoritmos e Estrutura de Dados II

Professor: André Vignatti

Resumo—O Trabalho tem o objetivo de implementar e analisar os principais algoritmos recursivos vistos em aula.

I. QUANTIDADE DE COMPARAÇÕES FEITAS ENTRE ELEMENTOS DO VETOR (NUMCOMP)

Figura 1: No Insertion Sort, o melhor caso é $\text{numComp} = n-1$ (vetor em ordem crescente), enquanto o pior caso é $\text{numComp} = n^2/2$ (vetor em ordem decrescente);

Figura 2: O Selection Sort possui apenas um caso, que é o caso médio: $\text{numComp} = n^2/2$;

Figura 3: O Merge Sort possui apenas um caso, que é o caso médio: $\text{numComp} = n(\log_2(n)+1)$;

Figura 4: No Quick Sort, o pior caso é $\text{numComp} = n^2/2$ e o melhor caso é $\text{numComp} = n \cdot \log_2(n)$;

Figura 5: No Heap Sort, o pior caso é $\text{numComp} = n \log_2(n) + 4n$.

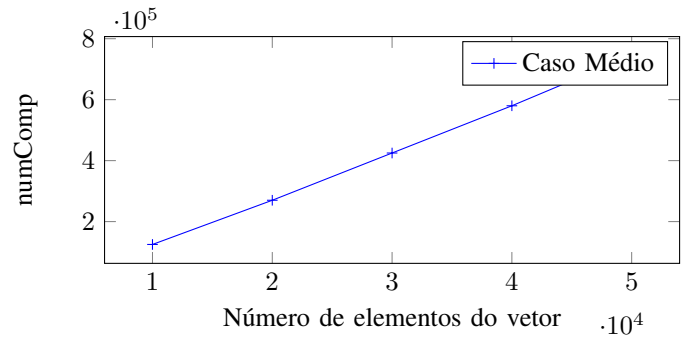


Figura 3. Gráfico do Merge Sort

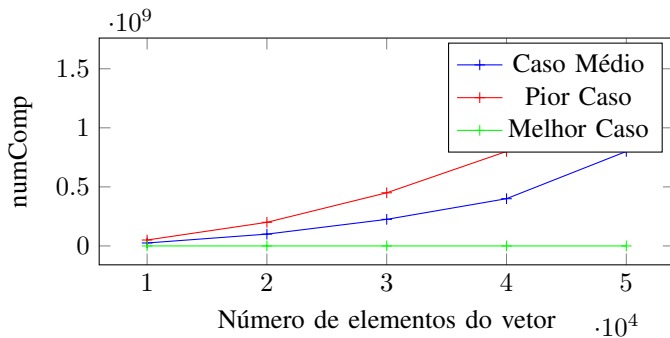


Figura 1. Gráfico do Insertion Sort

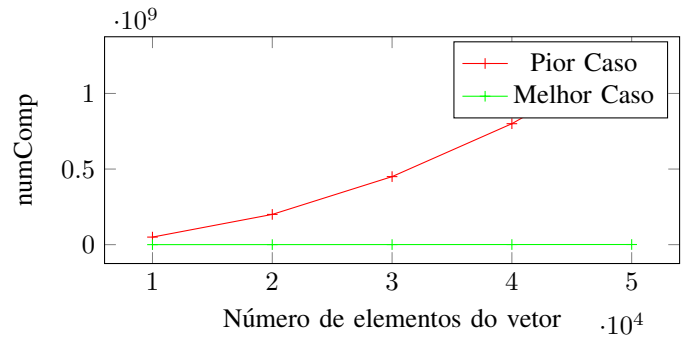


Figura 4. Gráfico do Quick Sort

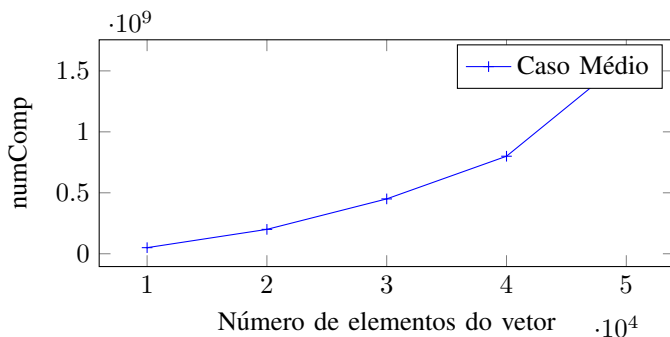


Figura 2. Gráfico do Selection Sort

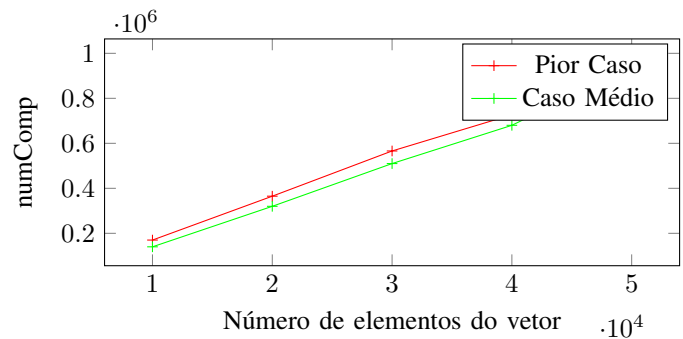


Figura 5. Gráfico do Heap Sort

II. TEMPO DE EXECUÇÃO DOS ALGORITMOS EM SEGUNDOS

Insertion Sort	10k	20k	30k	40k	50k
reverso	0.233s	0.598s	1.090s	2.032s	3.052s
random	0.062s	0.250s	0.555s	0.890s	1.476s
Selection Sort	10k	20k	30k	40k	50k
reverso	0.421s	1.87s	3.98s	6.935s	11.209s
random	0.421s	1.86s	3.95s	6.877s	10.614s
Merge Sort	10k	20k	30k	40k	50k
random	0.0009s	0.0019	0.0028	0.0039s	0.00507s
random	0.0009s	0.0031	0.0053	0.0079s	0.01032s
Quick Sort	10k	20k	30k	40k	50k
reverso	0.01828s	0.7094s	1.5697s	2.7544s	4.349s
random	0.00118s	0.0025s	0.0039s	0.005225s	0.005s
Heap Sort	10k	20k	30k	40k	50k
reverso	0.00198s	0.00399s	0.00670	0.01135s	0.0112s
random	0.002286s	0.00496s	0.01007	0.012994s	0.0178s

Figura 6. Tabela dos tempos em segundos para cada tamanho de vetor

Por meio dessa tabela, concluímos que os algoritmos mais eficientes são o Merge Sort, Quick Sort e o Heap Sort, pois eles apresentam, no geral, um tempo menor de ordenação. Além disso, vale ressaltar a congruência com os gráficos do número de comparações, visto que o Selection Sort e o Merge Sort, por apresentarem apenas casos médios, apresentam o mesmo tempo para o vetor desordenado, quanto para o vetor reverso (decrecente). Além disso, nota-se um atraso considerável no tempo dos algoritmos Insertion Sort e Quick Sort, quando testados com o vetor reverso. Por fim, nos testes realizados, o Merge Sort se mostrou como o mais rápido.

III. BUSCA SEQUENCIAL X BUSCA BINÁRIA

Para vetores ordenados, a Busca Binária é mais eficiente e possui $\text{numComp} = \log_2(n)$, enquanto a Busca Sequencial possui $\text{numComp} = n$, no pior caso. Dessa forma, a Busca Sequencial só se mostra interessante quando se trata de vetores não ordenados. Como exemplo, para um vetor com 50000 elementos, a Busca Sequencial levou 0.001130s para varrer o vetor inteiro, enquanto a Binária levou apenas 0.000002s.

IV. REFERÊNCIAS

- 1) "Vignatti, Andre. Video-Aulas.":