Universidade Federal da Fronteira Sul

Campus de Chapecó

Curso de Ciências da Computação

Patrícia Trevisan

Chapecó, 15 de novembro de 2020.

Análise de desempenho do método *Radix Sort*

Neste trabalho trago a análise de desempenho do método de ordenação *Radix Sort*, que nada mais é que um código capaz de ordenar vetores baseando-se nos dígitos dos seus elementos.

Foram utilizados nove vetores no total, com três tipos de ordenação diferentes, os que já estavam ordenados (V[n] = {1,2,3,…,n}), os em ordem decrescente (V[n] = {n, n-1,n-2,…,1}) e vetores em ordem aleatória, utilizando a ferramenta *rand* para números de 0 até 1000.

Ao analisar o gráfico e a tabela I percebemos que os vetores ordenados ou decrescentes mantém a ascendência da curva conforme o tamanho do vetor aumenta, porém quando comparamos com os vetores aleatórios um comportamento diferente se apresenta, ele se mantém estável no intervalo entre os tamanhos de 50 e 100 mil.

Esse comportamento deve-se ao fato de que os valores utilizados para gerar os vetores aleatórios não ultrapassam os 4 dígitos, podendo nem existir um elemento de 4 dígitos dependendo de como se comportar o *rand*, enquanto que os outros vetores chegam a casa dos 6 dígitos quando utilizamos o vetor com 100 mil elementos.

Pela análise feita podemos chegar a conclusão de que o que torna o método *Radix* mais rápido ou mais lento não é exatamente o tamanho ou a ordem dos elementos no vetor e sim o número de dígitos que contém os elementos.

Os testes foram rodados em uma máquina com processador *IntelCore i3 e 4*G de RAM.

Tabela I

Radix Sort	10000	50000	100000
Crescente	5.322 ms	14.822 ms	28.486 ms
Decrescente	5.571 ms	13.840 ms	26.671 ms
Aleatório	3.292 ms	13.614 ms	13.136 ms

Gráfico I