

Relatório algoritmos de ordenação Shell Sort e Selection Sort

Julia Rodrigues Gubolin

Docente: Aleardo Manacero Junior

Disciplina: Algoritmos e Técnicas de Programação II

Bacharelado em Ciência da Computação UNESP - IBILCE

24 de janeiro

Introdução

Neste relatório, será descrita a análise de tempo para ordenação dos algoritmos Shell Sort e Selection Sort.

Desenvolvimento

Para desenvolver a análise, foi usado o algoritmo Shell Sort de Knuth e o algoritmo Selection Sort. Como entrada, foi utilizado um arquivo com um conjunto de números formatados da maneira especificada no enunciado do projeto e gerados por um gerador de números randômicos online.

Foram utilizados 5 arquivos de entrada diferentes: o primeiro com 20 números aleatórios, o segundo com 200, o terceiro com 2000, o quarto com 20000 e o quinto com 200000. Estes números são importantes para a análise pois com o aumento deles é possível verificar se algum dos algoritmos apresentará um mal desempenho em detrimento do outro. Os valores destes arquivos foram colocados em um vetor com as respectivas posições e cada um deles foi executado pelo menos 5 vezes. Os resultados obtidos estão nas tabelas a baixo.

Tabela 1: Vetor de 20 posições.

Execução	Shell Sort	Selection Sort
1	0.000005	0.000006
2	0.000005	0.000005
3	0.000005	0.000006
4	0.000005	0.000006
5	0.000004	0.000005

Tabela 2: Vetor de 200 posições.

Execução	Shell Sort	Selection Sort
1	0.000070	0.000148
2	0.000074	0.000149
3	0.000070	0.000149
4	0.000071	0.000148
5	0.000071	0.000147

Tabela 3: Vetor de 2000 posições.

Execução	Shell Sort	Selection Sort
1	0.002744	0.005431
2	0.003168	0.011944
3	0.002984	0.006101
4	0.002533	0.005202
5	0.002628	0.005906

Tabela 4: Vetor de 20000 posições.

Execução	Shell Sort	Selection Sort
1	0.111506	0.478198
2	0.126695	0.629252
3	0.110199	0.548848
4	0.129836	0.486219
5	0.143153	0.482854

Tabela 5: Vetor de 200000 posições.

Execução	Shell Sort	Selection Sort
1	16.411950	123.342073
2	17.204910	124.469536
3	18.075372	123.531584
4	16.733274	123.875695
5	17.149721	122.035683

Conclusão

Com base nos dados recolhidos, pode-se concluir que o algoritmo com melhor desempenho é o Shell Sort, já que ele apresenta tempos de execução até 7 vezes menores que o Selection Sort, considerando o vetor de 200 mil posições.

Este resultado é esperado, pois em termos de complexidade Big O, o Shell Sort possui uma complexidade $\theta^{4/3}$ e o Selection Sort uma complexidade θ^{n^2} , o que mostra que de fato o algoritmo Shell Sort possui melhor desempenho. Além disso, no caso do Selection Sort, se o vetor dobra de tamanho, seu tempo de execução quadruplica, o que também explica os valores obtidos.