

TRABALHO PRATICO 2 - QUESTÃO 3

AVALIAÇÃO DO DESEMPENHO DOS ALGORITMOS DE ORDENAÇÃO EM 3 CASOS COM ENTRADA DE 1 MILHÃO DE NÚMEROS

- CASO 1 - COM NÚMEROS GERADOS ALEATORIAMENTE

Podemos perceber que o desempenho dos métodos BubbleSort e SelectionSort são semelhantes quando temos uma entrada bem grande (1 milhão de dados no caso) e números gerados aleatoriamente, ambos foram bem demorados passando de uma hora de duração. Além disso, o número de movimentações e comparações foram bem elevados. Ambos passaram de 1 bilhão e meio de comparações; já o número de movimentações do Bubble passou de 3 bilhões enquanto o do Selection passou de 1 bilhão.

O InsertionSort levou 18 minutos e meio para ser executado e passou de 1 bilhão de comparações e movimentações. Os métodos MergeSort, QuickSort e ShellSort levaram um tempo absurdamente menor do que todos os métodos anteriores, com todos levando menos de 1 segundo para serem executados. O número de comparações não passou de 25 milhões e o de movimentações não passa de 20 milhões, com exceção do Shell que faz aproximadamente 74 milhões.

É claro que quando se trata de uma entrada aleatória ela pode ou não favorecer algum método devido à disposição dos números.

- CASO 2 - COM NÚMEROS GERADOS EM ORDEM CRESCENTE

O desempenho de todos os métodos quando a entrada já estava ordenada foram bem rápidos com menos de 1 segundo de duração. A única exceção foi o SelectionSort que levou aproximadamente 30 minutos para ser executado, já que este método percorre todos os dados selecionando cada um e colocando na posição certa, mesmo que, no caso em questão, eles já estivessem nas posições certas.

É interessante observar que nesse caso os métodos Bubble e Insertion que tinham tido um desempenho pior no primeiro caso, nesse caso foram melhores, além de não fazerem movimentações. Os métodos Merge, Quick e Shell acabam fazendo movimentações mesmo com o vetor já ordenado, já que usam a estratégia de subdividir o vetor até o caso base para depois ordena-lo, usando a estratégia de dividir para conquistar.

- CASO 3 - COM NÚMEROS GERADOS EM ORDEM DECRESCENTE

Quando a entrada foi feita em ordem decrescente é nítido que realmente se trata do pior caso quando falamos do InsertionSort, visto que foi nesse caso que ele teve um maior tempo de execução.

Quanto aos demais métodos, Bubble e Selection assim como no caso da ordem aleatória foram os que levaram mais tempo, com ambos levando aproximadamente uma hora. Os métodos Merge, Quick e Shell novamente se destacaram pela rapidez levando menos de 1 segundo, com destaque para o Quick que manteve a média de 0,6 segundos em todos os 3 casos.

OBSERVAÇÃO: O tempo de todas as execuções é um fator que depende também da capacidade de cada computador, sendo assim ele pode variar de acordo com a máquina na qual for executado.