

Relatório - Trabalho 2 de Introdução à Ciência da Computação

Nome: Emanuel Carlos de A. Valente

Nº USP: 7143506

Física Computacional - 1º Semestre/2010

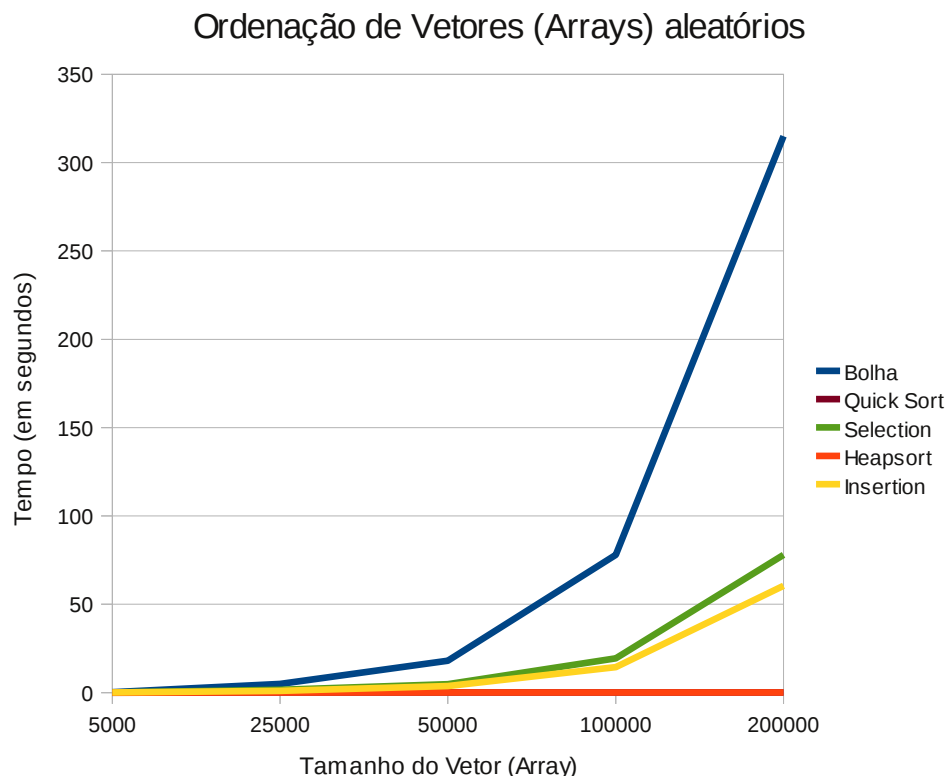
Prof. Dr. Alneu de Andrade Lopes

Foram implementados 5 tipos de algoritmos de ordenação: Bubble Sort, Insertion Sort, Selection Sort, Quick Sort e Heap Sort. Para testá-los foram criados vetores (arrays) de 5 mil, 25 mil, 50 mil, 100 mil e 200 mil posições. Cada teste mede o tempo, empiricamente, usando a função clock() (time.h), de cada ordenação dos algoritmos. O primeiro teste (T1) foi realizado com cada array desordenado, o segundo (T2) com elementos ordenados e o último teste com os elementos ordenados em ordem decrescente (T3).

As informações de hardware e software onde os testes foram realizados são: Computador Dell Optiplex 210L: Processador Pentium 4 HT 3.0 Ghz, Motherboard Intel, 1GB de memória DDR2, 80GB HD; GNU Linux Kubuntu 10.04 (kernel 2.6.32-22).

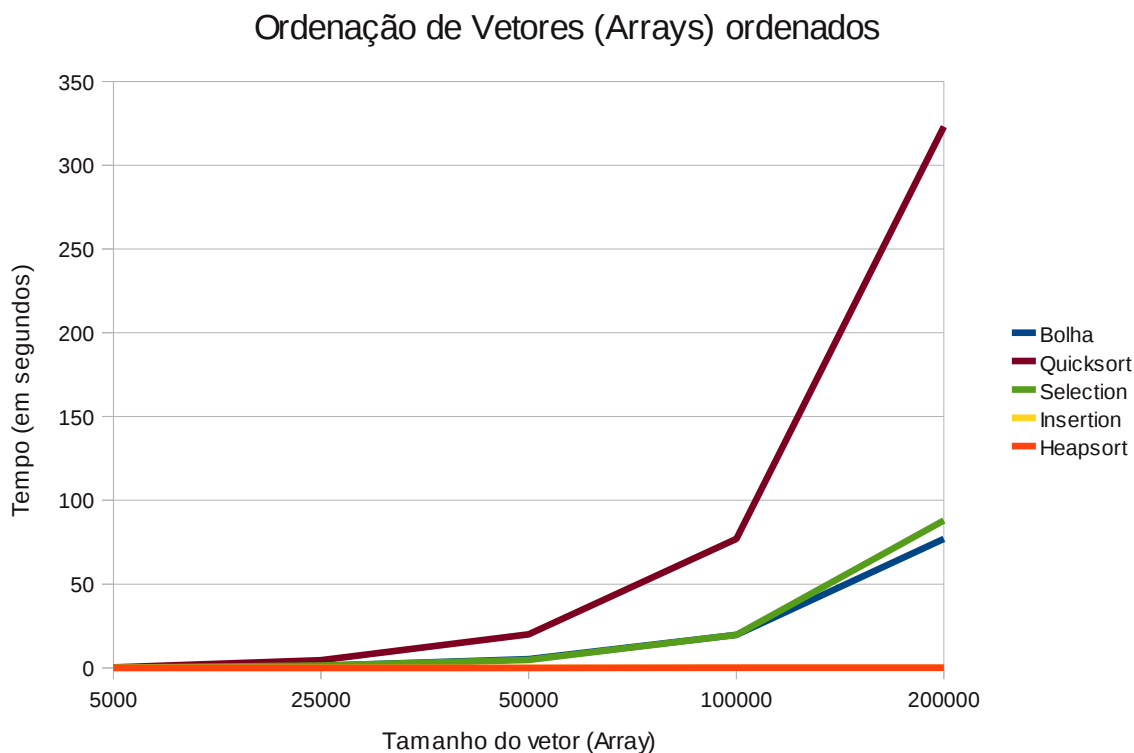
Abaixo estão os gráficos dos testes realizados no ambiente acima citado.

Primeiro Teste (T1)



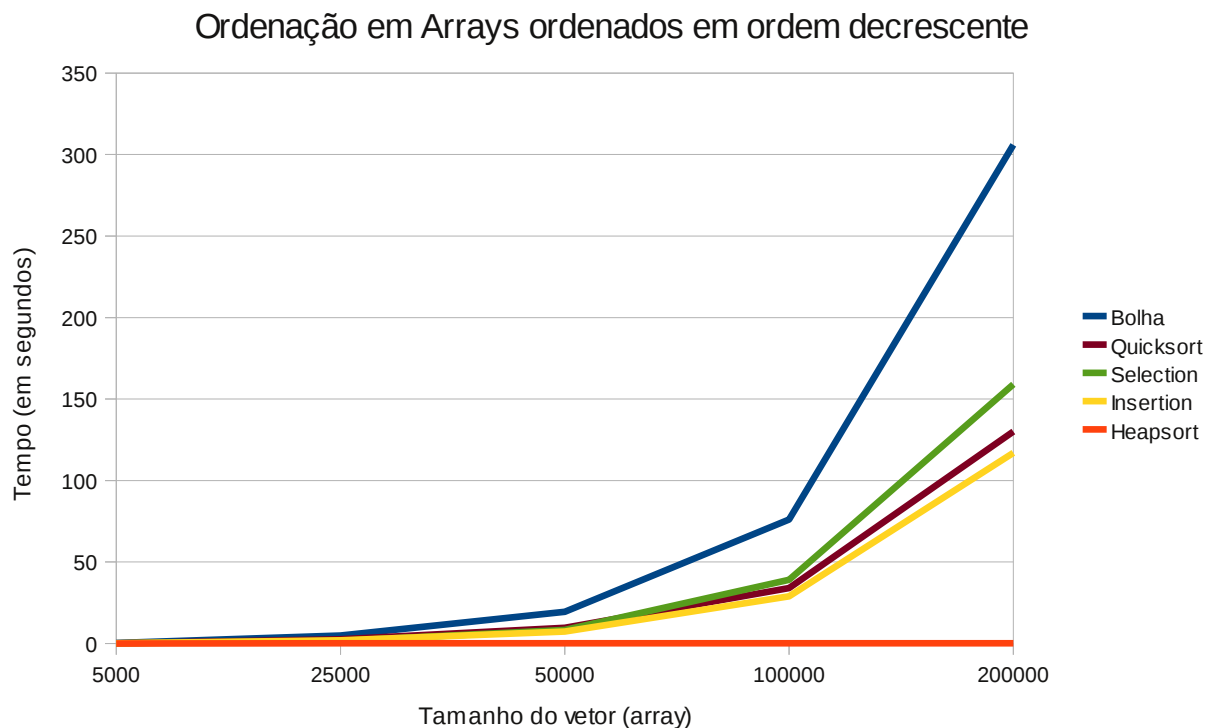
No primeiro teste o algoritmo Bubble Sort tem a menor performance, pois a média de sua complexidade é $O(n^2)$. Para este caso, os melhores algoritmos foram o Heap Sort e Quick Sort, com complexidade média de $O(n \log n)$. Note que a linha do Quick Sort no gráfico não é mostrada, pois sua performance é superior ao Heap Sort. O método Selection e Insertion são ambos de complexidade média de $O(n^2)$, porém um pouco mais sofisticados que o Bubble Sort.

Segundo Teste (T2):



O detalhe do segundo teste é que esse tipo de dados é o pior caso do Quick Sort, ou seja, seu nível de complexidade sai de $O(n \log n)$ para $O(n^2)$. Heap Sort se mantém em $O(n \log n)$. Nesse ambiente, Insertion Sort tem o seu melhor caso, $O(n)$. Os métodos Bubble e Selection tem sua complexidade mantida em $O(n^2)$.

Terceiro Teste (T3):



No último teste, concluímos que o Bubble Sort é o pior dos 5 aqui testados, pois seu nível de complexidade sempre será $O(n^2)$. Quick Sort e Insertion Sort apresentam neste caso complexidade $O(n)$, Selection Sort, em todos os casos $O(n^2)$, porém melhor que o Bubble. O Heap Sort sempre terá sua complexidade mantida em $O(n \log n)$, porém para conjuntos de dados aleatórios o Quick Sort é mais rápido.

O fato de Heap Sort ter sua complexidade de $O(n \log n)$ mantida não justifica que ele seja o melhor para ser usado em todos os casos. Como vimos aqui, o Quick Sort se mostrou mais ágil na ordenação com elementos aleatórios, porém sua performance não é a melhor com elementos ordenados. Todos esses algoritmos podem ser modificados de maneira que seu pior caso fique otimizado, como é o exemplo da função `qsort()`, disponível em `stdlib.h`, que é uma implementação de Quick Sort com modificações que melhoram o seu pior caso.