Comparando algoritmos de ordenação

Aluno: Jonatha Salles Menezes
Matrícula: 2211312125

Tecnologia em Análise e Desenvolvimento de Sistemas

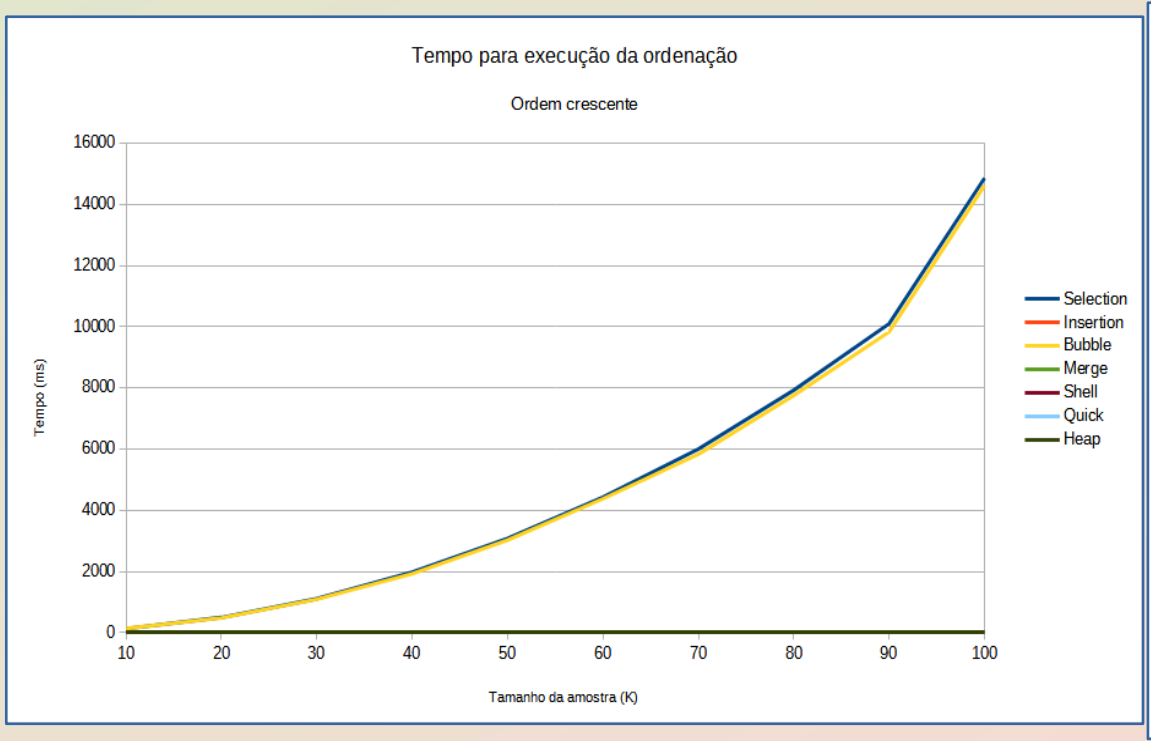
Estruturas de Dados II – CCB1050

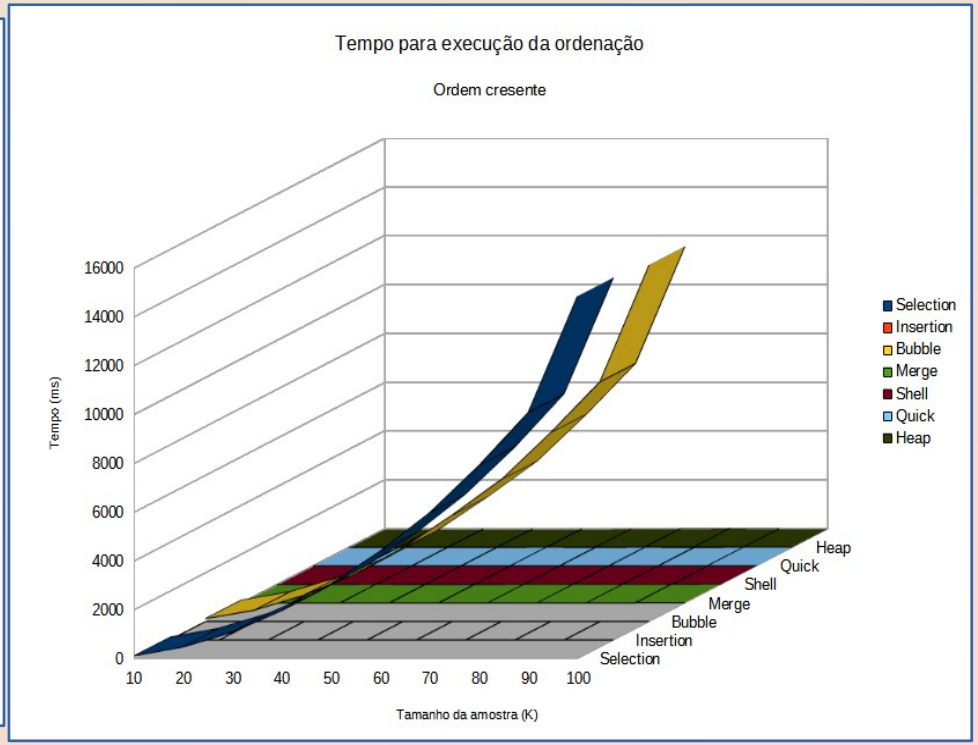
Prof. Eugênio Silva

Conteúdo

- Ordenação de sequências crescentes;
- Ordenação de sequências decrescentes;
- Ordenação de sequências randômicas (embaralhadas)

Sequências crescentes





Análise

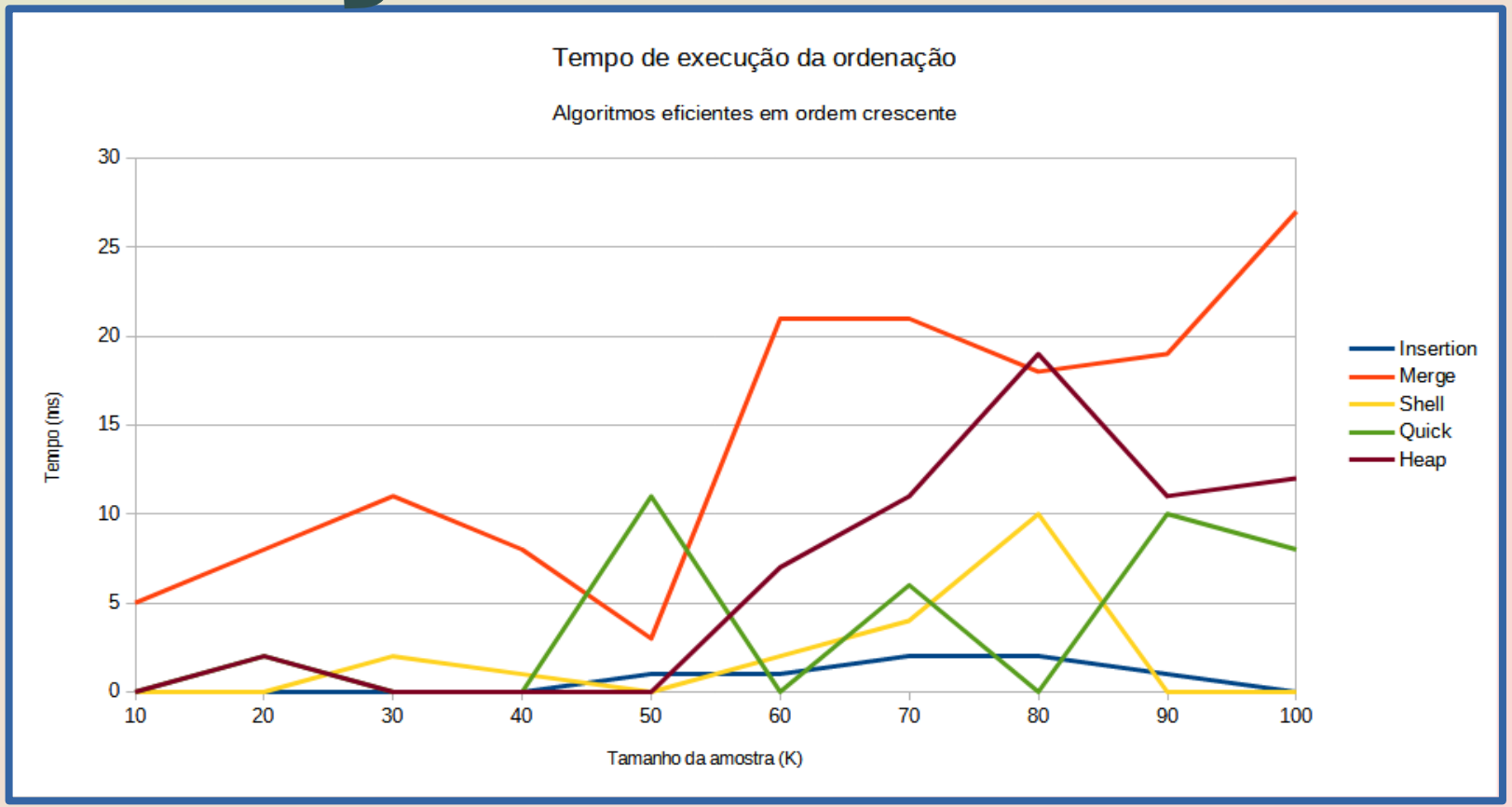
Olhando para o gráfico 2D da comparação, é possível determinar que:

- O algoritmo de seleção foi o mais demorado a partir da amostra de tamanho 60 mil.
- Antes de 60 mil, o algoritmo da bolha era o que levava mais tempo para ordenar.

Ao analisar o gráfico 3D, nota-se que:

- Os algoritmos eficientes e o algoritmo de inserção são os que levam menos tempo para ordenar. O tempo levado por esses algoritmos é de no máximo 27 milissegundos.
- A comparação entre esses algoritmos não é muito confiável pelo tempo tomado ser muito pequeno e variável, mas pode-se ter uma noção de comparação com a figura a seguir.

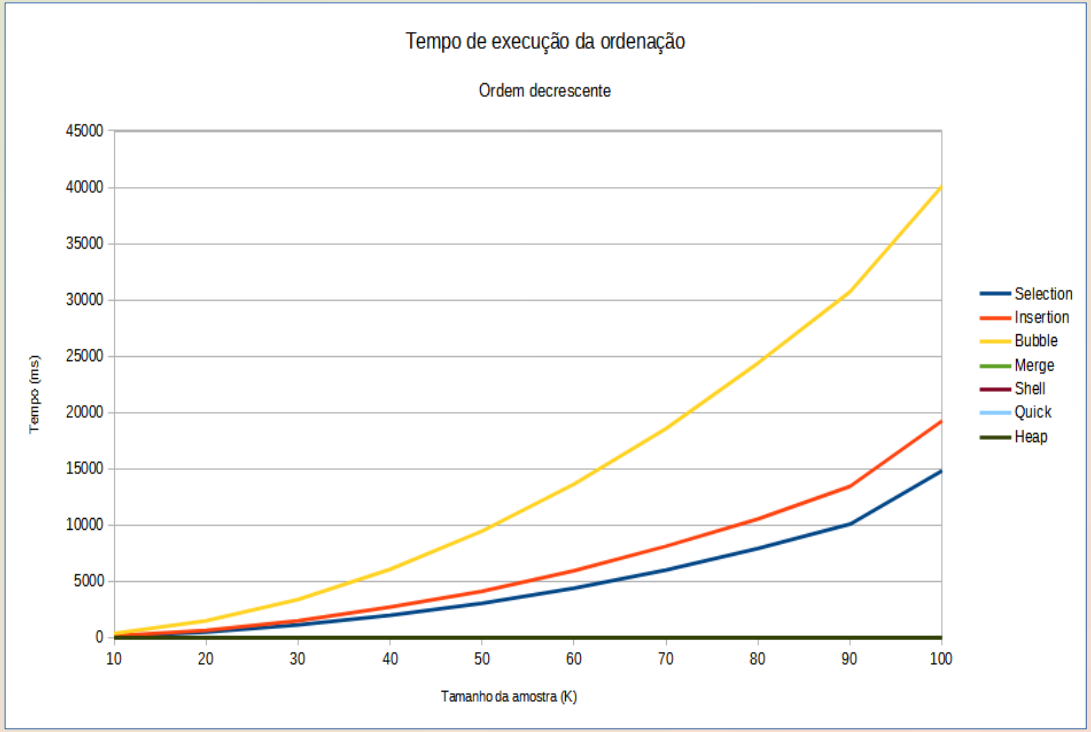
Comparando os algoritmos eficientes

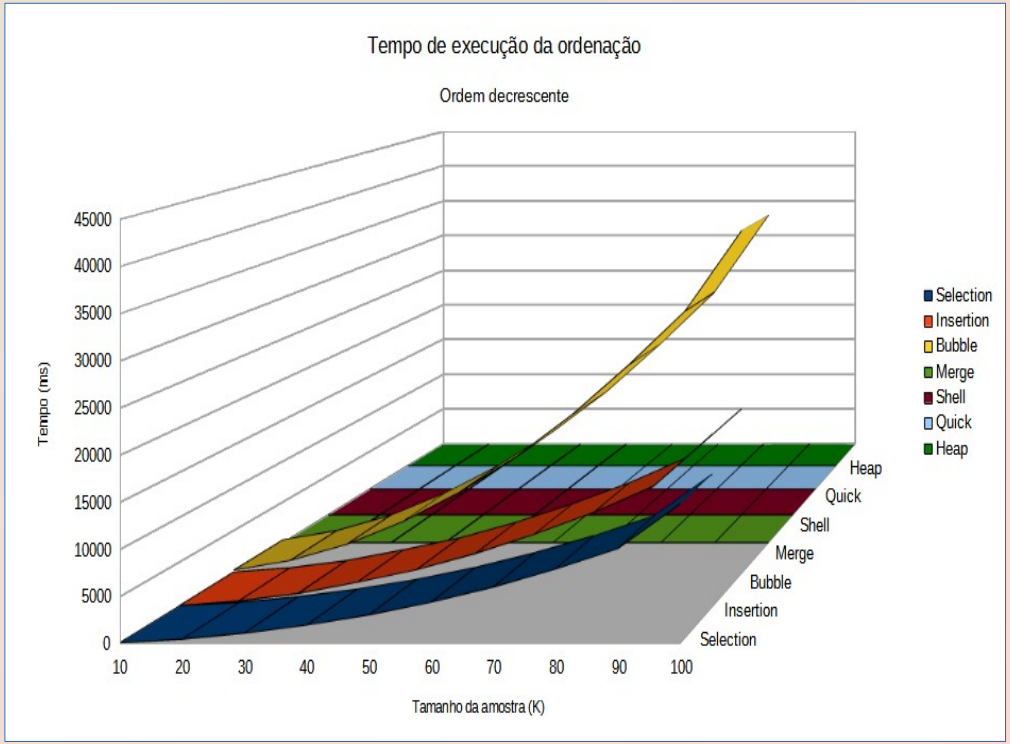


Análise

- Apesar de figurar entre os algoritmos simples, o de inserção é o mais eficiente para sequências já ordenadas.
- O MergeSort (ordenação por intercalação) parece ser o algoritmo eficiente mais lento para uma sequência já ordenada.
- O HeapSort e o QuickSort parecem ser o algoritmo mais oscilantes em relação ao tempo de execução entre os algoritmos eficientes.

Sequências decrescentes



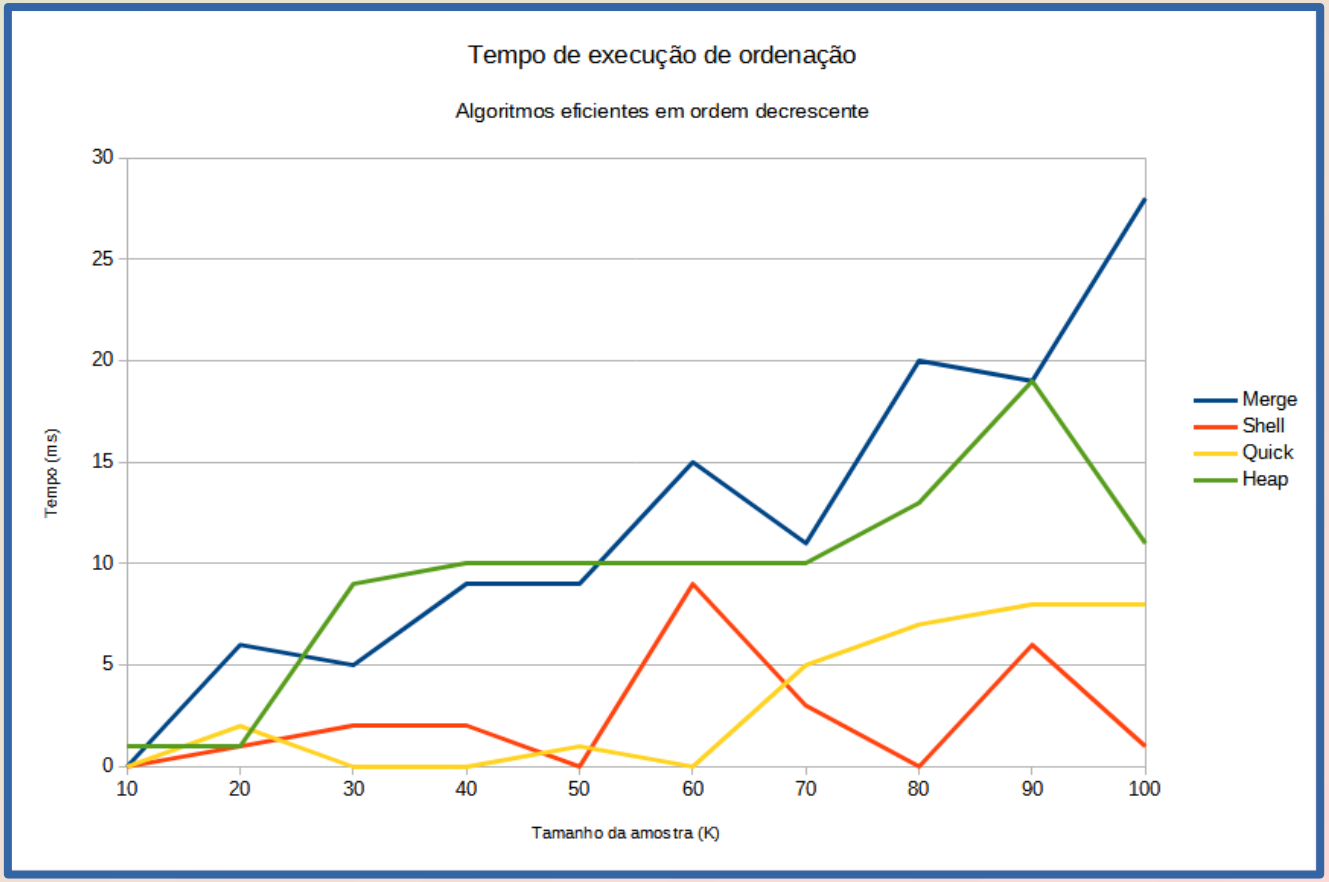


Análise

Ao analisar os gráficos, nota-se:

- O algoritmo da bolha é extremamente lento, levando até 40 segundos para uma ordenação.
- Que os outros algoritmos simples levam metade ou menos tempo que a bolha para executar a tarefa. A inserção seguida pela seleção.
- No gráfico 3D, novamente é possível notar que os algoritmos eficientes são muitas vezes mais rápidos que os algoritmos simples, por isso, iremos novamente precisar de um gráfico detalhado.

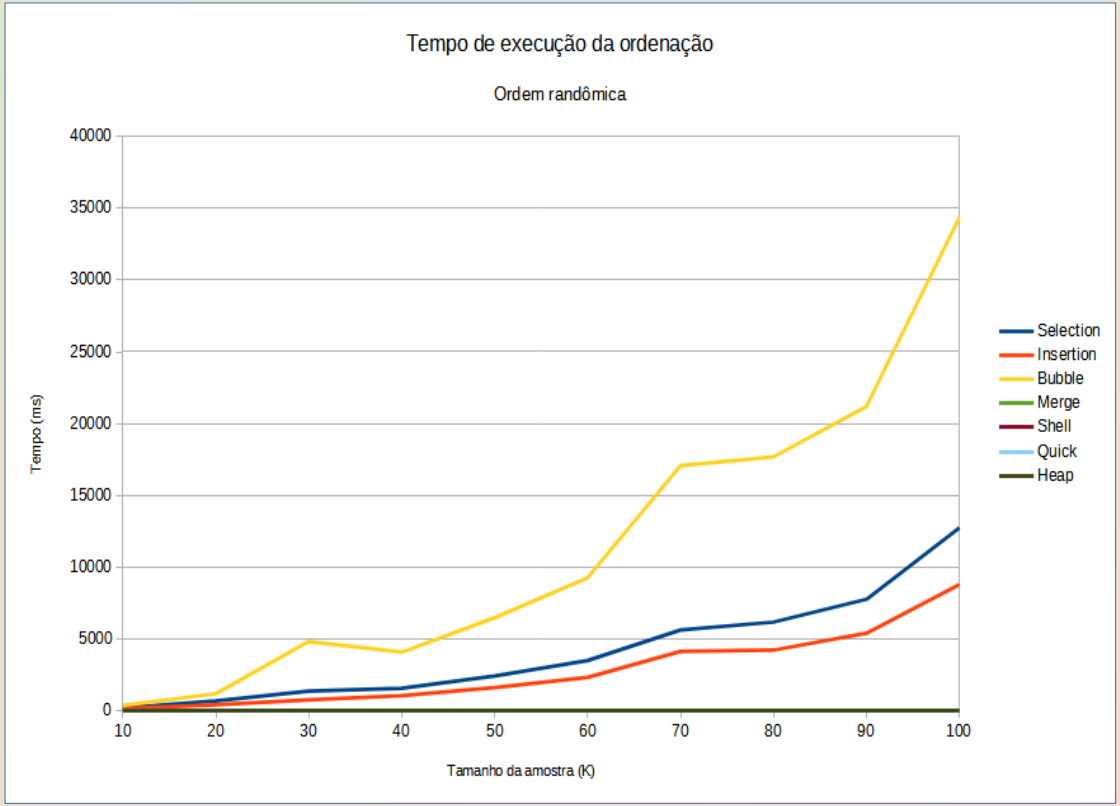
Comparando os algoritmos eficientes

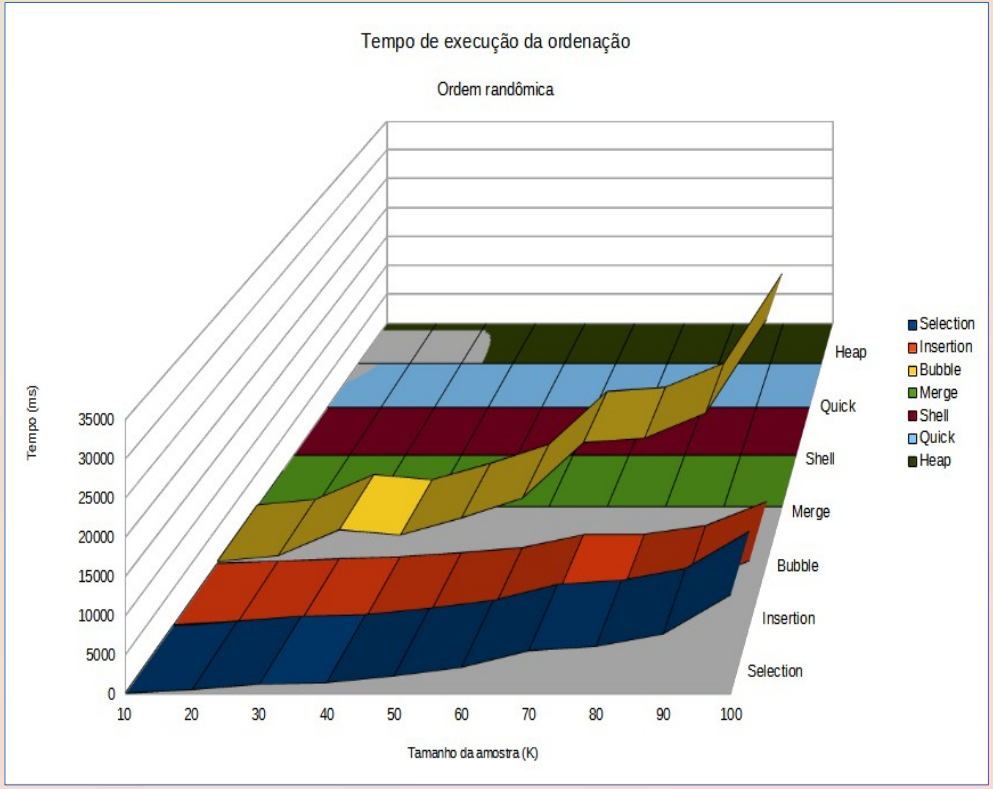


Análise

- Com a análise desse gráfico, é fácil inferir que os algoritmos QuickSort e ShellSort são os mais rápidos, pois consistentemente executam em menos tempo que os outros.
- O MergeSort parece ser o mais lento, seguido pelo HeapSort.

Sequências randômicas (embaralhadas)

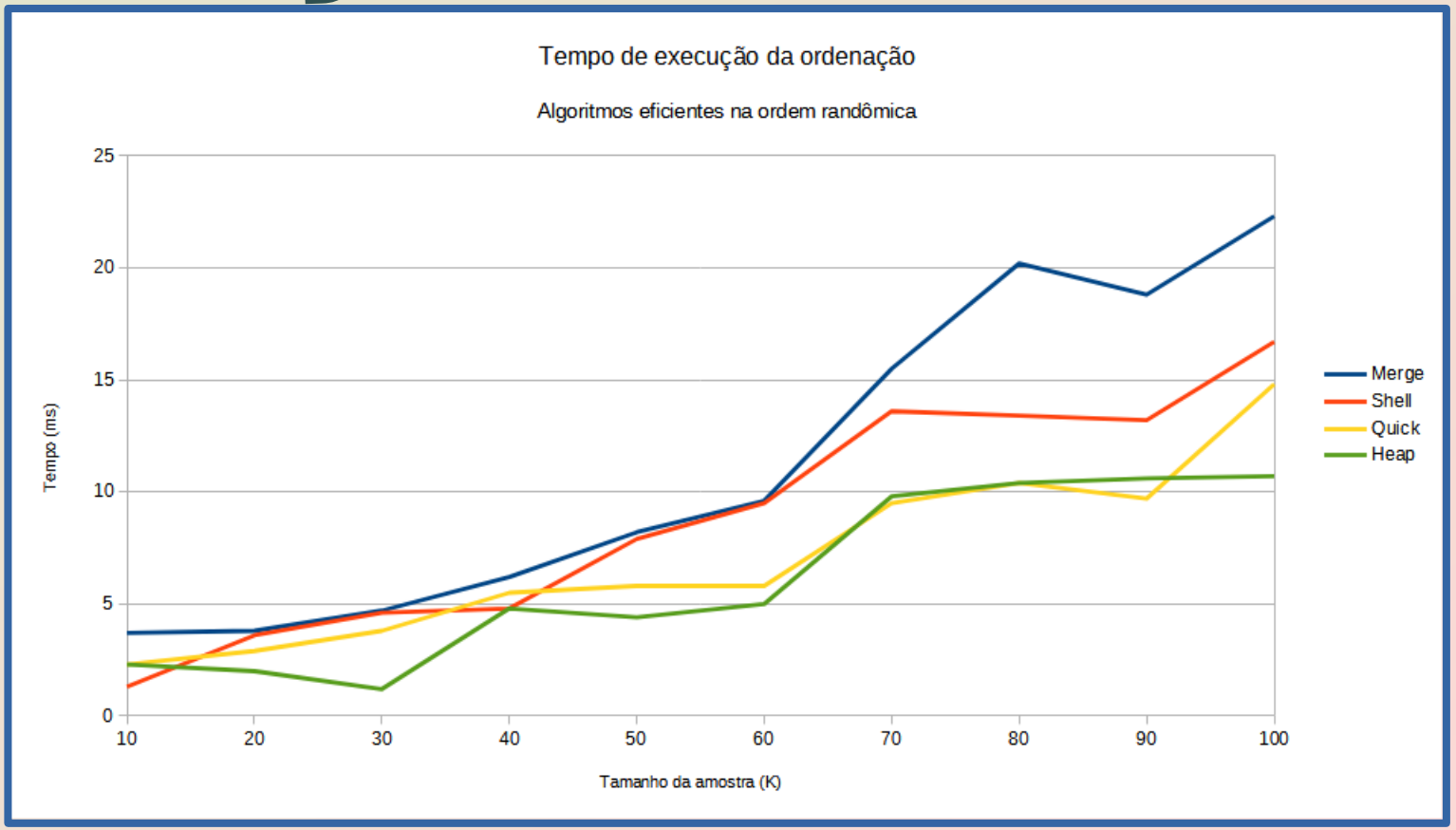




Análise

- Ao analisar os gráficos, nota-se:
- Novamente o algoritmo da bolha demorando várias vezes mais que os outros algoritmos simples.
- Seleção e inserção próximos outra vez, mas o algoritmo de inserção é mais rápido em todos os tamanhos da amostra.
- Por outra vez os algoritmos eficientes imensuráveis com os gráficos que incluem algoritmos simples.

Comparando os algoritmos eficientes



Análise

- A exemplo da ordenação inversa (decrescente), o padrão notado é o MergeSort sendo o algoritmo eficiente mais lento.
- No entanto, diferente do que se percebeu pela ordenação inversa, o HeapSort é o algoritmo mais rápido, junto ao QuickSort, na ordenação embaralhada.
- O ShellSort é mais rápido apenas que o MergeSort na ordenação embaralhada.

Considerações finais

- O algoritmo de inserção é um algoritmo simples, mas supera os algoritmos eficientes em sequências já ordenadas.
- O algoritmo de seleção é o algoritmo simples mais rápido a ordenar uma sequência inversa.
- O algoritmo da bolha é constantemente o mais lento e praticamente não possui pontos positivos.
- O QuickSort é o mais equilibrado, realiza os três tipos de ordenação com a mesma rapidez, praticamente.
- O MergeSort parece ser o algoritmo eficiente mais lento nos três tipos de ordenação.
- O ShellSort parece ser ligeiramente mais rápido que o QuickSort em sequências já ordenadas.
- O HeapSort tem eficiência semelhante ao QuickSort em sequências embaralhadas


UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
UERJ