

**BCC202 – Estrutura de Dados I**  
**2s2015**  
**Aula Prática 06**  
**Ordenação – ShellSort e HeapSort**

**Data de Entrega:** 10/02/2016 até 23h55 (via *Moodle*)

---

### **Questão 01**

Implemente o algoritmo de ordenação ShellSort usando três diferentes sequências para *h*. Cada variação receberá apenas um vetor de inteiros e o tamanho deste vetor e retornará o número de comparações de chaves e movimentos de registros realizados durante sua execução, além do tempo de execução. **Deixe como comentário no código fonte a escala de tempo escolhida (*mili segundos, nano segundos, minutos, ...*) e justifique sua escolha.**

Uma das sequências para *h* deve ser a de *Knuth*, vista em sala de aula, as outras duas devem ser escolhidas da *Wikipedia*: <http://en.wikipedia.org/wiki/Shellsort>. **No código fonte deixe comentários claros sobre quais foram as sequências para *h* escolhidas e onde elas foram implementadas.**

### **Exemplo de entrada e Saída**

A entrada inicia com o número de vetores a serem ordenados. Cada vetor é iniciado pelo número de elementos contidos no vetor e é seguido dos elementos a serem ordenados.

A saída apresenta a quantidade de comparações e movimentos realizados, seguidos pelo tempo de execução de cada algoritmo, seguindo o padrão definido no exemplo a seguir. O tempo de execução para os demais é opcional, mas para o ShellSort é obrigatório! Recomenda-se coletar o tempo de todos os algoritmos para comparação.

A saída deve incluir os resultados dos algoritmos implementados na *Prática 07*.

Entrada	Saída
2 10 1 2 3 4 5 6 7 8 9 10 10 5 4 3 2 1 6 7 8 9 10	Vetor 1: - Comparacoes   Movimentacoes   Tempo: - - Bubble....: C11   M11 - - Selection.: C12   M12 - - Insertion.: C13   M13 - - Merge.....: C14   M14 - - Quick.....: C15   M15 - - Shell 01...: C16   M16   T16 - - Shell 02...: C17   M17   T17 - - Shell 03...: C18   M18   T18  Vetor 2: - Comparacoes   Movimentacoes   Tempo: - - Bubble....: C21   M21 - - Selection.: C22   M22 - - Insertion.: C23   M23 - - Merge.....: C24   M24 - - Quick.....: C25   M25 - - Shell 01...: C26   M26   T26 - - Shell 02...: C27   M27   T27 - - Shell 03...: C28   M28   T28

Os valores Cxy, Mxy e Txy correspondem aos valores retornados pelas funções de ordenação para o número de comparações de chaves, o número de movimentações de registros e o tempo de ordenação, respectivamente. Tais valores não serão apresentados aqui, pois podem variar de uma implementação para outra.

DICA: Para que se consiga fazer comparações de tempo, é necessário fazer testes com vetores grandes. Crie um programa que gere a entrada recebendo como argumento o número de casos de teste e a dimensão de cada vetor. É interessante criar vetores ordenados de forma crescente, decrescente e randômicos para testar melhor, pior e caso médio.

## Questão 2

Implemente o algoritmo de ordenação HeapSort. Seu algoritmo receberá um vetor de inteiros e o tamanho deste vetor e retornará o número de comparações de chaves e movimentos de registros realizados durante sua execução, além do tempo de execução, da mesma forma como foi feito na Prática 08.

### Exemplo de entrada e Saída

A entrada inicia com o número de vetores a serem ordenados. Cada vetor é iniciado pelo número de elementos contidos no vetor e é seguido dos elementos a serem ordenados.

A saída apresenta a quantidade de comparações e movimentos realizados, seguidos pelo tempo de execução de cada algoritmo, seguindo o padrão definido no exemplo a seguir. Recomenda-se coletar o tempo de todos os algoritmos para comparação.

A saída deve incluir os resultados dos algoritmos de ordenação implementados nas práticas anteriores.

Entrada	Saida
2 10 1 2 3 4 5 6 7 8 9 10 10 5 4 3 2 1 6 7 8 9 10	Vetor 1: - Comparacoes   Movimentacoes   Tempo: - - Bubble.....: C11   M11 - - Selection.: C12   M12 - - Insertion.: C13   M13 - - Merge.....: C14   M14 - - Quick.....: C15   M15 - - Shell 01...: C16   M16   T16 - - Shell 02...: C17   M17   T17 - - Shell 03...: C18   M18   T18 - - Heap.....: C19   M19   T19  Vetor 2: - Comparacoes   Movimentacoes   Tempo: - - Bubble.....: C21   M21 - - Selection.: C22   M22 - - Insertion.: C23   M23 - - Merge.....: C24   M24 - - Quick.....: C25   M25 - - Shell 01...: C26   M26   T26 - - Shell 02...: C27   M27   T27 - - Shell 03...: C28   M28   T28 - - Heap.....: C29   M29   T29

Os valores Cxy, Mxy e Txy correspondem aos valores retornados pelas funções de ordenação para o número de comparações de chaves, o número de movimentações de registros e o tempo de ordenação, respectivamente. Tais valores não serão apresentados aqui, pois podem variar de uma implementação para outra.

DICA: Para que se consiga fazer comparações de tempo, é necessário fazer testes com vetores grandes. Crie um programa que gere a entrada recebendo como argumento o número de casos de teste e a dimensão de cada vetor. É interessante criar vetores ordenados de forma crescente, decrescente e randômicos para testar melhor, pior e caso médio.