

**TRABALHO 3: ALGORITMOS DE ORDENAÇÃO (ENTREGA EM 06/07/2017)**  
**LINGUAGEM C | COMPILADOR GCC | SISTEMA OPERACIONAL LINUX**

Considere os seguintes algoritmos de ordenação estudados:

1. Bubble-sort original;
2. Bubble-sort melhorado;
3. Insertion-sort;
4. Mergesort;
5. Quicksort com pivô sendo o último elemento;
6. Quicksort com pivô sendo um elemento aleatório;
7. Quicksort com pivô sendo a mediana de três;
8. Heapsort.

Faça um único programa que contenha:

- 1) Todos esses algoritmos implementados usando uma estrutura de dados do tipo vetor;
- 2) Os algoritmos 2, 3, 4 e 5 usando uma lista duplamente encadeada sem nó cabeça.

As seguintes características devem ser implementadas:

- Cada elemento é do tipo `int32_t`;
- A entrada será lida de um arquivo binário que contém uma sequência finita de elementos.
- O nome do arquivo binário de entrada deve ser passado como o primeiro argumento do programa;
- A saída – composta pela sequência ordenada dos elementos da entrada – deverá ser em um arquivo binário que contém o mesmo nome do arquivo de entrada seguido do sufixo `.out`.

Teste cada uma das versões dos algoritmos implementados com três tipos de entrada: em ordem crescente, em ordem decrescente e em ordem aleatória. Devem ser consideradas entradas dos seguintes tamanhos nos testes:

- 10.000, 100.000 e 500.000 (todas as versões de todos os algoritmos);
- 1.000.000, 10.000.000 e 100.000.000:
  - Mergesort, para vetor e lista encadeada;
  - Quicksort com pivô sendo um elemento aleatório;
  - Quicksort com pivô sendo a mediana de três;
  - Heapsort.

Faça um relatório com até 3 páginas contendo tabelas e discussões sobre os resultados obtidos nos testes. O relatório deve conter os dados do aluno e da disciplina, uma breve introdução, uma seção com os resultados obtidos e uma seção com a discussão sobre os resultados e conclusão do relatório. **Os tempos de execução devem ser coletados no Laboratório de Computação IV. O tempo deverá ser exibido em segundos, com precisão de 6 casas decimais (microsegundo).**

O código deve estar devidamente organizado, identado, comentado e sem *warnings* usando as opções -Wall -Wextra -pedantic do gcc.

**Cópias e similares: nota zero para todos os envolvidos.**