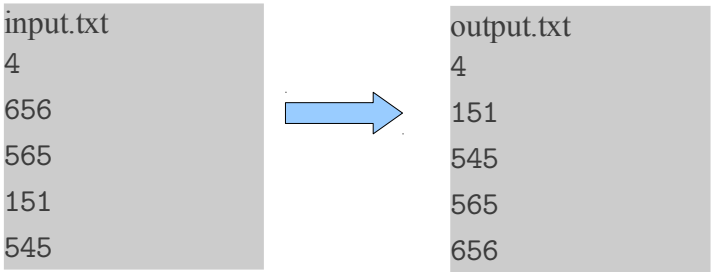
Trabalho Ordenação de Dados.

Instâncias de benchmark.

Serão fornecidas instâncias de benchmark que serão arquivos textos contendo números inteiros que deverão ser ordenados gerando (opcionalmente) um arquivo de saída com os elementos ordenados em ordem crescente.



Os arquivos estão divididos em duas classes, randômica e semi-ordenado. Cada classe possui 50 arquivos, sendo 10 arquivos de 5 tamanhos distintos por classe. Além disso a classe de semi-ordenados está subdividida em crescente e decrescente.

Equipes.

Serão permitidos equipes de até três alunos

Nota.

A nota decorrerá da apresentação do trabalho no labcomp, do domínio demonstrado pelos componentes das equipes sobre sua implementação, pela qualidade do material escrito e relevâncias dos comentários.

Entrega.

Data de entra: dia 17 no labComp.

Tarefas.

1 – Implemente os seguintes algoritmos de ordenação:

a) Selection Sort
b) Insection Sort
c) Bubble Sort
d) Shell Sort
e) Merge Sort
f) Quick Sort
g) Heap Sort

2 – Implemente uma versão do Quick Sort na qual um subconjunto com menos de k elementos seja ordenado por um procedimento de Bubble Sort interrompendo a chamada recursiva. Teste para k = 10, 20, 40, 80.

3 – Utilize uma função de ordenação oferecida pela API da linguagem selecionada para ordenar os arquivos de benchmark.

4 – Aplique sua implementação aos arquivos de Benchmark e escreva um relatório sobre o comportamento dos mesmos e comente sobre os resultados.