Trabalho em Grupo 1
Algoritmos de Ordenação

Escolha pelo menos um algoritmo de cada um dos três grupos: I={Bubble, Selection, Insertion}, II={Mergesort, Heapsort, Quicksort, Shell}, III={Counting Sort, Radix Sort, Bucket Sort} que leia da entrada *N* entradas e compare os tempos de execução para diferentes cenários.

Escreva cada algoritmo em um programa separado e em uma ou mais linguagens de programação (C, C++, Java ou Python). A quantidade de linguagens de programação deve ser equivalente à quantidade de alunos no grupo.

Cenários: seu grupo deve escolher os cenários a serem testados pelo seu grupo, sendo que cada cenário deve ser testado com entradas de diferentes (pelo menos 5) tamanhos *N* (1.000 <= *N* <= 1.000.000). A quantidade de cenários deve ser maior ou igual à quantidade de alunos no grupo.

a) Números inteiros/reais aleatórios
b) Números inteiros/reais em ordem não-ascendente
c) Números inteiros/reais quase ordenados (>= 95% de valores em ordem)
d) Cadeias de caracteres (ex: nomes de pessoas) em ordem aleatória
e) Cadeias de caracteres (ex: nomes de pessoas) quase ordenados (>= 95% de valores em ordem)
f) Ordenação por múltiplos critérios (ex: dados com múltiplos atributos)
g) Outro a ser proposto

O grupo deverá criar seus conjuntos de dados e/ou obter os dados a serem ordenados a partir de dados públicos de websites (ex: openml http://www.openml.org). Para conjuntos de dados com mais de uma classe (*class*), você pode dividir os dados por classe. A ordenação deve ordenar as instâncias (*instance*) por meio de um ou mais atributos (*features*), definindo-se uma ordem de prioridade entre os atributos. O seu programa deve ler essa entrada (formato csv, por exemplo) e criar uma tabela com os dados a serem ordenados.

Execute as suas implementações de algoritmos de ordenação para cada arquivo de entrada, calculando os tempos de execução de cada programa para cada entrada. Faça três execuções para cada combinação e calcule o seu tempo de execução médio. Ao se calcular o tempo de execução, não faça impressão na saída.

Faça um relatório contendo gráficos que mostram comparativos de desempenhos de cada algoritmo/implementação para cada tipo de entrada. Crie gráficos que descrevem os perfis (média, mediana, quartil superior e inferior, etc) dos dados utilizados também. Descreva os detalhes dos experimentos realizados (geração dos arquivos de entrada, ambiente computacional utilizado nos experimentos, etc) e discuta sobre os resultados observados.

Formato de entrega

Arquivo comprimido contendo os seguintes arquivos:

i) Relatório em PDF ou Google Docs

ii) Códigos de cada implementação

Tamanho dos Grupos

Até 3 alunos (de preferência em grupo).