 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA</p> <p><i>Departamento de Engenharia Informática</i></p>	<p>Trabalho Prático nº 4 - “Algoritmos de Ordenamento”</p> <p>Algoritmos e Estruturas de Dados</p> <p>2015/2016 – 2º Semestre</p> <p>Tarefa A: 29/Abril/2016, 23h00 Tarefa B: 15/Maio/2016, 23h00</p> <p>Relatório: 16/Maio/2016, 17h00 <i>Em papel, manuscrito, e entregue na secretaria do DEI</i></p>
--	--

Objectivos: Implementação e estudo de algoritmos de ordenamento

Notas Importantes:

- O relatório do trabalho deve ser MANUSCRITO, no formulário anexo a esta ficha.
- Para o aluno ser avaliado neste trabalho deve assegurar cumulativamente: (1) entrega do relatório do trabalho e submissão do código no mooshak nos prazos estabelecidos; (2) ter no mínimo DUAS presenças nas aulas PL em que está inscrito e dedicadas a este trabalho.

Introdução do Problema

Muitos problemas em geometria computacional são caracterizados por terem um conjunto de pontos num espaço d -dimensional como dados de entrada. Tipicamente, os primeiros passos na resolução deste tipo de problemas consiste na ordenação dos pontos de entrada para serem posteriormente processados de forma eficiente. O objectivo deste trabalho é ordenar, por ordem lexicográfica, um conjunto de n pontos d -dimensionais usando algoritmos de ordenamento. Para simplificar, neste trabalho vamos assumir que as coordenadas dos pontos são inteiras.

Problema A

Dados n pontos d -dimensionais, imprimir os pontos por ordem lexicográfica crescente, ou seja, devem ser ordenados pela primeira coordenada e em caso de empate, devem ser ordenados pela segunda coordenada, em caso de empate, ordenados pela terceira, etc.

Descrição da entrada:

- 1ª linha: Contém dois números inteiros, n e d , separados por espaços
- As n linhas seguintes contêm, cada uma, d valores inteiros entre -999 e 999, separados por espaços

n – número de pontos ($2 \leq n \leq 1\,000\,000$)

d – número de dimensões ($1 \leq d \leq 3$)

Descrição da Saída:

- **n** linhas, cada linha contém **d** valores inteiros, separados por espaços

Entrada do Exemplo 1:

```
4 2
105 145
5 202
245 202
5 102
```

Saída do Exemplo 1:

```
5 10
5 202
105 145
245 202
```

Entrada do Exemplo 2:

```
5 1
82
56
1
510
54
```

Saída do Exemplo 2:

```
1
54
56
82
510
```

Tarefas do Problema

Tarefa A:

1. Implementar o algoritmo de ordenamento elementar **insertion sort**.
2. Implementar um algoritmo eficiente de ordenamento por comparação de chaves.


Tarefa B:

3. Implementar um algoritmo de ordenamento por comparação de radicais.

Nota: A escolha e implementação dos algoritmos deve ter em conta a eficiência temporal e espacial dos mesmos.

Estudo Experimental

Tarefa C: Medir e analisar os tempos das soluções implementadas e escrever o relatório em anexo.

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA</p> <p><i>Departamento de Engenharia Informática</i></p>	<p>Trabalho Prático nº 4 - “Algoritmos de Ordenamento”</p> <p>Algoritmos e Estruturas de Dados</p> <p>2015/2016 – 2º Semestre</p> <p>Tarefa A: 29/Abril/2016, 23h00 Tarefa B: 15/Maio/2016, 23h00</p> <p>Relatório: 16/Maio/2016, 17h00 <i>Em papel, manuscrito, e entregue na secretaria do DEI</i></p>
--	--

Nome: _____ nº _____
Mooshak login: _____

Estudo Experimental

De entre os seguintes conjuntos de pontos:

- pontos por ordem ascendente
- pontos por ordem descendente
- muitos pontos repetidos
- pontos muito desordenados (gerados aleatoriamente)
- outro

Deverá escolher 3 conjuntos que sejam os mais adequados para cada um dos algoritmos implementados, e que devem representar:

1. um conjunto fácil
2. um conjunto difícil
3. um conjunto intermédio

Deverá gerar estes conjuntos de dados, considerando $d=3$. O terceiro conjunto deverá, obrigatoriamente, corresponder ao conjunto de pontos gerados aleatoriamente.

Apresente numa folha em anexo:

- I. Gráfico com os tempos de execução da solução implementada para o algoritmo elementar da Tarefa A para os 3 conjuntos referidos acima
- II. Gráfico com os tempos de execução da solução implementada para o algoritmo eficiente da Tarefa A para os 3 conjuntos referidos acima
- III. Gráfico com os tempos de execução da solução implementada para a Tarefa B para os 3 conjuntos referidos acima
- IV. Gráfico com os tempos de execução da *solução eficiente* implementada para a Tarefas A e da solução implementada para a Tarefa B, para o conjuntos de entrada 3.

Notas para construção dos gráficos:

1. Medir os tempos de ordenação para conjuntos com os seguintes tamanhos: 100, 200, ..., 1000, 2000, ..., 10000, 20000, ..., 100000, 200000, ..., 1000000
2. Os tempos medidos devem excluir o tempo despendido na leitura e na escrita dos dados.

3. Para cada tamanho da entrada, fazer 30 repetições e apresentar a média.
4. Para o Gráfico IV, os mesmos conjuntos de dados devem ser usados nas medições dos tempos dos dois algoritmos implementados.

Análise dos Resultados

1. Tarefa A – Algoritmo Eficiente

- 1.1. Qual o algoritmo eficiente de ordenamento que escolheu implementar? Porque motivo escolheu esse algoritmo?

- 1.2. Que estruturas de dados usou?

- 1.3. Qual a complexidade espacial requerida pela implementação? Justifique sucintamente.

- 1.4. Indique, justificando, que conjuntos de pontos construiu para os conjuntos de entrada 1 e 2.

1.5. Comente, sucintamente, se os resultados observados estão de acordo com o esperado.

1.6. Comparando os dois algoritmos implementados na Tarefa A (Gráficos I e II), que conclusões tira?

2. Tarefa B

2.1. Qual o algoritmo de ordenamento que escolheu implementar? Porque motivo escolheu esse algoritmo?

2.2. Que estruturas de dados usou?

2.3. Qual a complexidade espacial requerida pela implementação? Justifique sucintamente.

2.4. Indique, justificando, que conjuntos de pontos construiu para os conjuntos de entrada 1 e 2.

2.5. Comente, sucintamente, se os resultados observados estão de acordo com o esperado.

3. Comparação (Gráfico IV)

3.1. Compare os tempos de execução entre o algoritmo eficiente implementado para a Tarefa A e o algoritmo implementado para a Tarefa B. Comente e justifique as semelhanças/diferenças observadas.
