



**DEEC**

DEPARTAMENTO DE ENGENHARIA  
ELETROTÉCNICA E DE COMPUTADORES

TÉCNICO LISBOA

# Algoritmos e Estruturas de Dados

Licenciatura em Engenharia Electrónica (LEE)

2021/2022 (2º Sem. - P4)

Enunciado do Projecto

Ver. 1.0 (20220514)

## **CLASSIFICAÇÃO DE EMPRESAS**

Prof. Paulo Flores

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>O Problema - Classificação de Empresas</b>	<b>3</b>
<b>3</b>	<b>Especificação do Programa</b>	<b>4</b>
<b>4</b>	<b>Características do Programa</b>	<b>4</b>
4.1	Dados de entrada . . . . .	4
4.2	Dados de Saída - 1º Projecto . . . . .	4
4.3	Dados de Saída - 2º Projecto . . . . .	5
4.4	Execução do programa . . . . .	5
<b>5</b>	<b>Validador de resultados</b>	<b>6</b>
<b>6</b>	<b>Entrega do Projecto</b>	<b>6</b>
<b>7</b>	<b>Relatório Final</b>	<b>7</b>
<b>8</b>	<b>Avaliação do Projecto</b>	<b>8</b>
<b>9</b>	<b>Considerações Adicionais</b>	<b>8</b>

## Revisões

Ver.	Data	Autor	Descrição
1.0	2022-05-14	Paulo Flores	Versão inicial

# 1 Introdução

O trabalho que se descreve neste enunciado correspondem à avaliação do projecto da disciplina de Algoritmos e Estrutura de Dados da Licenciatura em Engenharia Electrónica. A descrição geral do trabalho corresponde ao problema a resolver para a avaliação no projecto. O primeiro projecto é baseado no mesmo problema mas consiste em obter alguns resultados, que apesar de não determinarem a solução do problema final, podem ser usados posteriormente para ajudar na sua resolução.

A entrega do código fonte de ambos os projectos é feita através de um sistema automático de submissão que verificará a sua correcção e testará a execução do programa em algumas instâncias do problema.

## 2 O Problema - Classificação de Empresas

Pretende-se neste trabalho desenvolver um programa que recebendo uma lista não ordenada de valores de facturação de empresas em milhões de euros, determine os valores limites e total de facturação, assim com a lista ordenada das empresas que estão dentro de um dado intervalo de categorias (*rankings*).

Ou seja, a partir de uma listagem de  $N$  empresas com os respectivos valores de facturação e de dois valores de *ranking*,  $P$  e  $Q$ , pretende-se obter de forma eficiente o valor de facturação mínimo, máximo e total das empresas dentro *ranking* (entre  $P$  e  $Q$ ), assim como a respectiva lista de empresas e facturação ordenada alfabeticamente. Define-se que a empresa com maior facturação será a primeira do *ranking* com valor igual a 1.

Na Figura 1 apresentar-se um exemplo de um problema de classificação de empresas com dez empresas ( $N = 10$ ), para dois valores de *ranking* de  $P = 3$  e  $Q = 7$ . Note-se que a ordenação dos dados para obtenção do *ranking* é apresentada apenas para efeitos de explicação do problema, mas não tem que necessariamente ser realizada explicitamente.

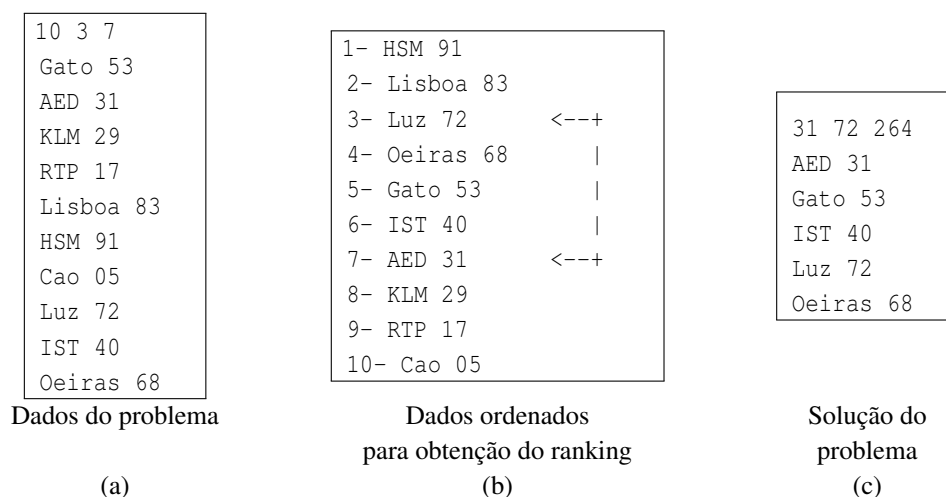


Figura 1: (a) Exemplo de um problema de classificação de empresas com  $N = 10$ ,  $P = 3$  e  $Q = 7$ ; (b) passo intermédio para explicar a solução; (c) a respectiva solução.

### 3 Especificação do Programa

O programa a desenvolver no projecto da disciplina lê os dados de entrada de um ficheiro com: o tamanho da lista de empresas ( $N$ ); os valores de  $P$  e  $Q$  do *ranking*; e a sequência de  $N$  empresas e respectivas facturações, com uma empresa por linha.

O algoritmo a implementar deve encontrar as empresas cujos valores de facturação estejam compreendidas entre a posição  $P$  e  $Q$  do *ranking*, e determinar o valor mínimo, máximo e total de facturação dessas empresas. Os valores obtidos, assim como a lista de todas essas empresas (dentro do *ranking*) e respectivas facturações ordenadas alfabeticamente, devem ser escritos num ficheiro de saída. Caso existam mais que uma empresa com a mesma facturação, as suas ordens relativas no *ranking* não são importantes.

Dado tratar-se de uma resolução computacional para o problema, pretende-se que a obtenção dos valores pedidos, seja feita com o menor dispêndio de recursos possível. Desta forma, serão valorizados os programas que encontrem mais rapidamente uma solução para o problema exigindo menos recursos.

### 4 Características do Programa

#### 4.1 Dados de entrada

O programa deverá ler um ficheiro com a descrição do problema (de extensão **.prb**) contendo os seguintes dados: o número total de empresas a considerar, os valores de  $P$  e  $Q$  e a listagem das empresas constituído por um código alfa-numérico de identificação da empresa e respectiva facturação.

O ficheiro **.prb** a ser lido com o problema tem as seguintes características:

- Cada ficheiro contém a descrição de apenas um problema de Classificação de Empresas.
- A 1ª linha do ficheiro tem três números inteiros,  $N$ ,  $P$  e  $Q$  separados por espaços. Estes números indicam respectivamente, o número de empresas contidas no ficheiro, e os dois valores do *ranking*.
- Cada uma das  $N$  linhas seguintes do ficheiro contém um código alfa-numérico (sem espaços), que identifica a empresa, e um inteiro que representa o respectivo valor de facturação. O identificador e o valor de facturação estão separados por espaços e/ou Tabs.

A título de exemplo, a Figura 1(a) mostra um ficheiro de um problema de classificação de empresas com 10 empresas (**prob00.prb**).

Na secção dedicada ao projecto na página da disciplina pode obter um ficheiro **zip** contendo alguns exemplos de problemas simples de classificação de empresas.

#### 4.2 Dados de Saída - 1º Projecto

O resultado de execução do programa para o 1º projecto resume-se apenas a calcular, o valor mínimo, máximo e total de facturação das empresas contidas no *ranking*, e os identificadores das empresas e facturação nas posições  $P$  e  $Q$  do *ranking*.

Nesta fase do projecto sugere-se aos alunos que usem explicitamente um algoritmo de ordenação para ordenar os valores de facturação (deve ser adaptado um dos algoritmos estudados na disciplina que seja mais adequado ao problema, podendo mesmo ser usada uma implementação da biblioteca do C).

Os resultados devem ser escritos, num ficheiro com o mesmo nome do problema mas com a extensão alterada para **.sol**. No total devem ser escritos na primeira linha 3 inteiros separados por espaços, correspondentes respectivamente ao valor mínimo, máximo e total de facturação das empresas contidas no *ranking*. Seguido de duas linhas com o identificador e facturação das empresas que estão na posição  $P$  e  $Q$ , respectivamente.

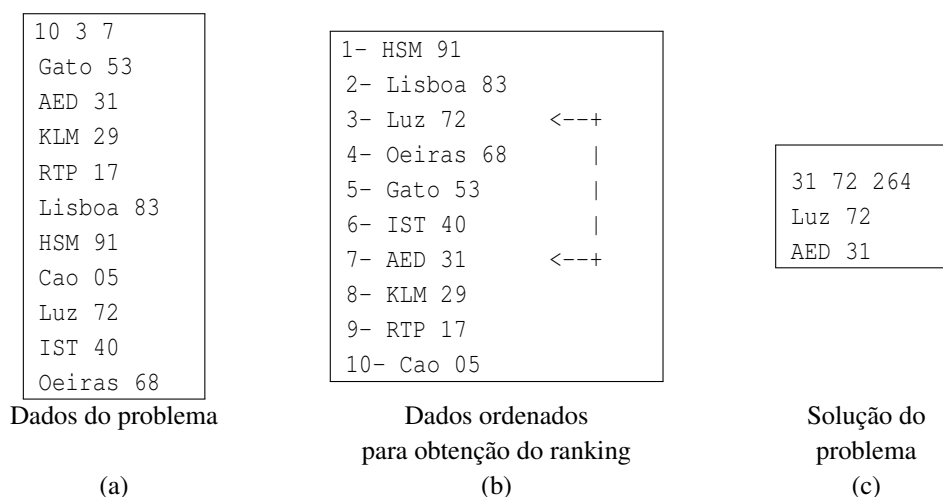


Figura 2: (a) Exemplo de um problema de classificação de empresas com  $N = 10$ ,  $P = 3$  e  $Q = 7$ ; (b) passo intermédio para explicar a solução; (c) a respectiva solução para o 1º projecto.

### 4.3 Dados de Saída - 2º Projecto

O resultado da execução do programa para o projecto deve determinar uma solução para o problema, ou seja, qual o valor da facturação máxima, mínima e total das empresas dentro do *ranking*  $P$  e  $Q$  e a lista das empresas e respectivas facturações ordenadas alfabeticamente.

A solução do problema deve ser escrita num ficheiro com o mesmo nome do problema mas com a extensão alterada para **.sol.i**. Neste ficheiro devem existir apenas  $Q - P + 2$  linhas. A primeira linha deve conter 3 inteiros (separados por espaços) que correspondem, respectivamente, à facturação mínima, máxima e total das empresas dentro *ranking*. Nas restantes linhas, cada linha deve ter o identificador da empresa e a sua facturação. Sendo que a listagem das empresas do *ranking* resultante deve ser apresentada ordenada alfabeticamente pelo identificador da empresa (de acordo com o código ASCII).

**Nota:** Espera-se que o desenvolvimento do algoritmo para o 2º projecto seja feito de forma que em média a resolução do problema seja mais eficiente que a implementação do 1º projecto, recorrendo para isso a técnicas que, por exemplo, não necessitam de ordenar todos os valores de facturação para resolver o problema.

### 4.4 Execução do programa

O programa desenvolvido para resolver o problema de classificação de empresas deverá ser invocado na linha de comandos da seguinte forma:

```
aed$ ./pce prob00.prb
```

onde:

**pce** designa o nome do ficheiro executável do programa (Projecto Classificação de Empresas);

**prob00.prb** é o nome do ficheiro com a descrição do problema de classificação de empresas, de acordo com o formato indicado na Secção 4.1.

Como resultado da execução do programa deve ser gerado um ficheiro com o nome **prob00.sol** ou **prob00.sol.i** com a solução do problema lido e no formato indicado na Secção 4.2 ou Secção 4.3 consoante se trate do 1º ou 2º projecto, respectivamente.

Assume-se que os formatos dos ficheiros contendo a descrição do problema estão sempre correctos pelo que o programa a desenvolver não tem de se preocupar com a detecção de erros na leitura dos dados de entrada.

**Nota:** a função **main()** deve terminar retornando obrigatoriamente o valor 0 (zero) no caso de execução do programa com sucesso.

## 5 Validador de resultados

Será disponibilizado um validador de resultados que permite verificar se um dado ficheiro solução apresenta uma solução correcta do problema de classificação de empresas. Eventualmente, o validador poderá ser noutra linguagem que não em C.

O modo de invocação do validador é um dos seguintes, de acordo se trate do 1º ou 2º projecto:

```
aed$ ./val1 prob00.prb prob00.sol
aed$ ./val2 prob00.prb prob00.sol1
```

onde:

**val1** ou **val2** designa o nome do ficheiro executável contendo o programa validador da solução para cada um dos projectos;

**prob00.prb** é o nome do ficheiro contendo a descrição do problema de classificação de empresas, conforme o formato indicado na Secção 4.1;

**prob00.sol** ou **prob00.sol1** é o nome do ficheiro contendo a solução gerada pelo programa desenvolvido no projecto nos formatos apresentados na secções 4.2 e 4.3, respectivamente.

Um ficheiro zip contendo o validador (e eventuais ficheiros de suporte) pode ser obtido na secção dedicada ao projecto na página da disciplina.

## 6 Entrega do Projecto

A entrega de ambos os projectos é feita de forma electrónica acedendo a um sistema automático de submissão e verificação. A submissão do código do projecto deverá respeitar o procedimento seguinte (instruções mais detalhadas podem ser encontradas na secção de projectos da página da disciplina):

- Na secção do projecto da página da disciplina deverá aceder à página indicada (site) para entrega de projectos e escolher o projecto respectivo (1º ou 2º projecto).
- Deverá efectuar a submissão (*upload*) de um ficheiro arquivo com extensão .zip (ou .7z ou .tgz) o qual deverá incluir o seguinte:
  - Os ficheiros fonte e respectivos cabeçalhos (ficheiros \*.c e \*.h), que constituem o programa e que são necessários para a geração do respectivo executável.
  - Um ficheiro **Makefile** que permita criar o executável na directoria de topo com o nome **pce** como resultado de executar o comando **make**.  
O compilador utilizado é o compilador gcc da GNU, devendo ser utilizadas as seguintes opções: **-Wall -ansi -pedantic**. A compilação não deve gerar erros/avisos com estas opções de compilação.
- Como resultado do processo de submissão será informado se a resolução entregue apresenta a resposta esperada num conjunto de casos de teste.

- Podem ser feitas múltiplas submissões sendo considerada como final a última versão a ser submetida. No entanto, o mesmo ficheiro (com o mesmo conteúdo zip, 7z, etc) não pode ser submetido mais do que uma vez.

**Nota:** cada grupo só pode fazer submissões com um mínimo de 10 minutos de intervalo.

Os prazos para submissão dos projectos electronicamente são apresentados na Tabela 1:

Tabela 1: Datas de submissão dos projectos.

Trabalho	Data limite de submissão	Relatório (em papel) entregue ao Docente
1º Projecto	até às 23h59 de 3 de Junho	Questionário fornecido pelo docente.
2ª Projecto	até às 23h59 de 24 de Junho	Relatório com o máximo de 5 páginas de texto e questionário fornecido pelo docente.

## 7 Relatório Final

O relatório final do projecto está limitado a 5 páginas de texto e deverá contemplar os aspectos seguidamente indicados relativos apenas ao projecto 2. A apresentação do mesmo deverá iniciar-se por aspectos de ordem geral, seguindo-se descrições mais detalhadas dos módulos e estruturas de dados utilizados. Sugere-se então a seguinte estrutura para o relatório:

- uma capa com os dados dos membros do grupo, incluindo nome, número. Esta capa deverá seguir o formato indicado na página da disciplina dedicada ao projecto (a capa não conta para o limite de páginas);
- uma descrição do problema que foi resolvido, com indicação clara das especificações do mesmo tal como foram entendidas;
- texto simples que indique como os alunos abordaram e resolveram o problema da classificação de empresas.
- uma descrição completa da arquitectura do programa, incluindo um fluxograma elucidativo e um texto claro mas sucinto indicando a divisão lógica e funcional dos subsistemas desenvolvidos para a resolução do problema, indicando os respectivos objectivos, as funções utilizadas e as estruturas de dados de suporte;
- uma descrição *detalhada* das estruturas de dados utilizadas e justificação para as mesmas;
- descrição dos algoritmos usados (por exemplo, na manipulação das estruturas de dados);
- uma análise dos requisitos computacionais do programa desenvolvido, tanto em termos da memória utilizada como da complexidade computacional, com particular ênfase no custo das operações de processamento sobre as estruturas de dados;
- pelo menos um exemplo de aplicação, o resultado obtido, e um ou mais exemplos significativos que mostrem os recursos usados em termos de tempo de computação e de memória. Em particular, a comparação dos tempos de execução para problemas grandes usando o programa/algoritmo do 1º e do 2º projecto (sugere-se o uso do comando linux **time**).

Não deve ser entregue qualquer código em papel, mas o código submetido deve ser estruturado de forma lógica em vários ficheiros (\*.c e \*.h). As funções devem ter um cabeçalho curto mas explicativo e o código deve estar correctamente indentado e com comentários de forma que facilitem a sua legibilidade.

**NOTA:** O relatório deve ser entregue ao docente (no gabinete 2.N-20) impresso em papel durante a 2ªf seguinte à data de submissão do projecto (27 de Junho).

## 8 Avaliação do Projecto

A avaliação do projecto é baseada em várias componentes:

- os algoritmos e as estruturas de dados usadas para resolver o problema;
- o desempenho da funcionalidade do programa realizado;
- a qualidade do código entregue (implementação, organização, indentação, comentários, etc);
- o relatório a descrever/documentar o projecto realizado.

Para a avaliação final poderão ser usados outros ficheiros de teste (maiores e/ou mais complexos) que não tenham sido usados para validar a entrega do projectos.

A nota final do projecto estará limitada a um valor máximo de acordo com o exposto na Tabela 2. Na avaliação de cada um dos projectos será também tido em conta se o programa apenas funciona parcialmente, embora o faça de forma correcta, ou se funciona de forma ineficiente, por exemplo faz uma má gestão de memória ou é ineficiente computacionalmente de uma forma grosseira. Serão valorizados os programas que funcionem de forma eficiente e em que o código está correctamente comentado e estruturado.

Tabela 2: Tabela de Valorização dos Projectos.

Situação	Nota Máx.
Programa não compila, a execução termina sistematicamente de forma ilegal ou com erros de execução ( <i>crasha</i> ), etc.	Trabalho não é aceite
Programa funciona de forma algo incompleta ou incorrecta, ou seja, determina alguns valores mas apresenta resultados total ou parcialmente errados, não obtém uma solução completa para o problema ou tem qualquer outra limitação considerada grave .	12 valores
Programa determina a solução do problema mas funciona de forma ineficiente, por exemplo, faz uma má gestão de memória ou é computacionalmente ineficaz.	16 valores
Programa funciona de forma correcta e usa eficientemente todos os recursos computacionais, com código correctamente comentado e estruturado e o relatório está bem escrito, claro e completo.	20 valores

## 9 Considerações Adicionais

Espera-se que os alunos conheçam e respeitem o **Código de Honestidade Académica** que rege esta disciplina e que foi apresentado no início do ano. Lembramos igualmente que a avaliação inclui a verificação de potenciais violações a este código. A detecção de situações de cópia entre projectos é feita de forma automática com recurso a sofisticados métodos de comparação de código, que envolvem não apenas a comparação directa do código mas também da sua estrutura.