

# **Estruturas de Dados e Algoritmos**

## **Avançados**

(ano letivo 2017-18)

### **e-fólio B**

Este enunciado constitui o elemento de avaliação designado por “e-fólio B” no âmbito da avaliação contínua e tem a cotação total de 4 valores. A sua resolução deve ser entregue até às 23h55 do dia 4 de dezembro pelos alunos que escolheram a modalidade de avaliação contínua.

A resolução deve ser entregue através de um único ficheiro compactado .zip, que:

- (i) contém os ficheiros .c ou .cpp que constituem o código dos programas, prontos a serem compilados;
- (ii) contém um ficheiro pdf de formato livre, com um relatório sucinto com informações complementares de modo a permitir uma fácil compreensão do trabalho realizado. É desnecessário incluir uma listagem integral do código.
- (iii) O nome do ficheiro .zip a entregar deve seguir a seguinte convenção para o seu nome,

“NumeroAluno-PrimeiroNome-Apelido-21045-efB.zip”

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro, “327555-Paulo-Costa-21045-efB.zip”

O ficheiro deve ser única e exclusivamente entregue através do recurso “E-fólio B” disponibilizado na plataforma (Nota: apenas é visível para os alunos inscritos em avaliação contínua), não sendo aceites trabalhos enviados por outras vias, como por exemplo por e-mail.

Esta é uma prova de avaliação **individual** e não “um trabalho de grupo”. A sua resolução deve provir unicamente do conhecimento adquirido e trabalho original desenvolvido pelo próprio aluno. Os alunos deverão saber distinguir claramente entre discutir os conteúdos abordados na unidade curricular (permitido) e discutir a resolução específica do e-fólio (não permitido).

## I

1. Escreva um programa em linguagem C ou C++ padrão, de nome `huff`, que calcule a codificação de Huffman para os símbolos (bytes) de um ficheiro.

- O programa deve aceitar como único argumento o nome de um ficheiro e escrever na saída padrão uma tabela com o seguinte formato,

```
37
0x38 0.002 0100011
0x32 0.01  00100
...
```

onde a primeira linha contém o número N de símbolos distintos codificados (símbolos que ocorreram pelo menos uma vez no ficheiro), seguido de N linhas, uma por símbolo, com o código hexadecimal do símbolo, a respetiva probabilidade de ocorrência e palavra de codificação de Huffman. A ordem das linhas deve refletir a ordem das folhas da árvore de Huffman da “esquerda para a direita”.

- O ficheiro de entrada só deve ser lido uma única vez e a leitura efetuada um byte de cada vez. Em caso algum o ficheiro de entrada deve ser todo lido previamente para memória.

- Admitindo “>> ” como prompt para a linha de comandos do sistema operativo, um exemplo de utilização do programa é,

```
>> huff ficheiro >tabelahuff.txt
```

Notas sobre o algoritmo (para normalização da codificação obtida):

(i) No início (e só no início), os símbolos são ordenados da “esquerda para a direita” por probabilidade. Símbolos com igual probabilidade são ordenados por código.

(ii) Na união de dois símbolos/árvores, a posição da união é a da esquerda.

(iii) Na união de dois símbolos/árvores quando existe mais do que uma possibilidade, são escolhidos os dois símbolos/árvores “mais à esquerda”.

(iv) Na codificação da palavra, é atribuído 0 ao ramo da esquerda e 1 ao da direita.

- O programa deve estar identificado com um cabeçalho similar ao seguinte,

```
/*
** UC: 21045-Estruturas de Dados e Algoritmos Avançados
** e-fólio B 2017-18 (huff)
**
** Aluno: 327555 - Paulo Costa
*/
```

- No desenvolvimento do programa em C++ não deve ser utilizada a STL. Restrições aplicam-se nomeadamente aos includes `<array>` `<deque>` `<forward_list>` `<list>` `<map>` `<queue>` `<set>` `<stack>` `<unordered_map>` `<unordered_set>` `<vector>` e em parte de `<algorithm>`. Em caso de dúvida questionar o seu uso. Não existem restrições para `<string>`.

**Critérios de correção:**

- Programa não compila com o compilador `gcc` ou `g++` => 0 valores.
- Programa não contém os respetivos `#include` para cada função ou classe que utiliza => 0 valores.
- Código do programa não está correta e uniformemente indentado de modo a permitir a sua leitura fácil => 0 valores
- Programa não está comentado => 0 valores. Os comentários no programa elucidam questões relevantes do código locais ao comentário.
- Programa não cumpre as especificações de entrada de dados => 0 valores na componente de funcionalidade, não será possível testar o programa.
- Programa não cumpre as especificações de saída de dados, não gerando os dados solicitados ou gerando dados a mais não solicitados => 0 valores na componente de funcionalidade, não será possível testar o programa. (sugestão: utilize uma flag em todos os prints que efetuar de dados não solicitados, entregue o programa com flag=0)
- O relatório deve dar uma boa ideia do trabalho efetuado sem ser necessário ver o código. Explique o como e porquê relativamente às opções e soluções técnicas que tomou para a estrutura e funcionamento do programa (até 35%).
- Funcionalidade do programa de acordo com o pedido, estrutura, nível de simplicidade e qualidade do código (até 65%)

**Nota ética:** Nunca é de mais referir que o código a apresentar como solução para este e-fólio deve ser 100% original do aluno. A probabilidade de duas pessoas que efetivamente não comunicaram entre si, apresentarem programas “quase iguais” é considerada nula. Isto é válido para qualquer par de alunos (cópia), assim como entre um aluno e qualquer outra pessoa, em particular através da Internet (cópia/plágio), onde existem inúmeras soluções e código para os mais variados problemas, em sites, fóruns, blogs, etc.

Cumpra estritamente as normas de realização individual, como se estivesse num exame com consulta, onde pode consultar a documentação mas não pode falar com ninguém.

FIM