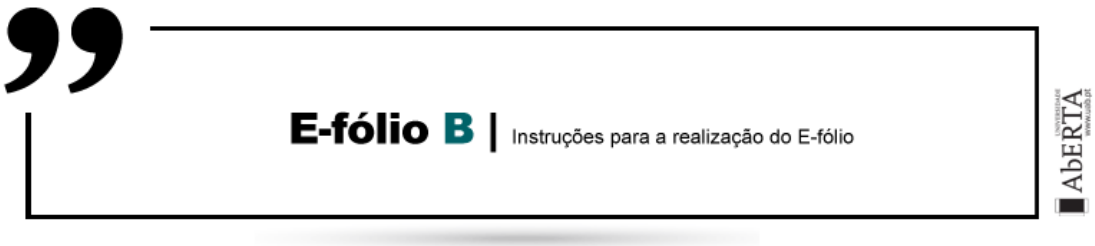
UNIVERSIDADE ABERTA www.uab.pt

Código 21046

Estruturas de Dados e Algoritmos Fundamentais

(ano letivo 2021-22)



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 20 de maio pelos alunos que escolheram a modalidade de avaliação contínua.

A resolução deve ser entregue através de um relatório constituído por um único ficheiro pdf, que deve seguir a seguinte convenção para o seu nome,

"NumeroAluno-PrimeiroNome-Apelido-21046-efB.pdf"

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro, "327555-Paulo-Costa-21046-efB.pdf" (sem acentos).

O ficheiro deve ser única e exclusivamente entregue através do recurso "E-fólio B" disponibilizado na plataforma moodle (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).

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.

No caso de dúvidas de interpretação do enunciado, utilize o fórum de avaliação para pedidos de esclarecimento.

I

1. Pretende-se desenvolver um programa em linguagem C++ padrão que implemente uma árvore binária de pesquisa (ABP), que contém informação sobre alunos, nomeadamente o seu número de aluno (que é a chave utilizada na ABP) e o seu nome. A implementação da ABP deve utilizar, de forma dinâmica, a estrutura fornecida e deve conter funções que permitam inserir alunos, retirá-los da árvore, procurar um número de aluno na árvore, encontrar o aluno com menor número e imprimir os alunos por ordem crescente do seu número, utilizando o formato: "Número de aluno, Nome" com um aluno por linha.

1.1 [2.5] Implemente a árvore binária de pesquisa, utilizando a estrutura dinâmica fornecida, na atividade "e-fólio B – VPL", disponível na página da unidade curricular. Deve escrever o seu código de forma a completar o ficheiro ABP.cpp. São fornecidos os ficheiros Aluno.h, e ABP.h, que não deve alterar. É ainda fornecido um ficheiro main.cpp que pode (e deve) alterar para testar exaustivamente o seu código (código que não compila no VPL será penalizado). Apresenta-se de seguida o output esperado para o ficheiro main fornecido:

```
Alunos ordenados:
500, Ana
1000, André
1500, Maria
2500, Cláudia
3000, Daniel

Encontrou Aluno: 1500, Maria

Aluno com menor número de aluno: 500, Ana
```

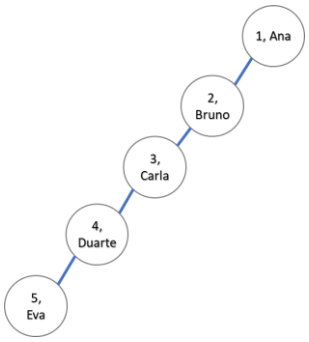
Critérios:

Construtor – 0.2 val.
Inserir – 0.5 val.
Apagar – 0.8 val.
Procura – 0.3 val.
Menor – 0.3 val.
ImprimeOrdenado – 0.3 val.
Destrutor – 0.1 val.

- Para cada função, desconto até 100% da cotação se:
    - implementam outro tipo de estrutura do que aquela indicada.
    - for grosseiramente ineficaz
    - não está comentado, é confuso e tem problemas com a indentação

1.2 [0.75] Indique qual é a complexidade, na notação Big-O, no pior caso possível, para as operações de inserção, procura e remoção numa árvore binária de pesquisa. Justifique e dê um exemplo em que tal aconteça. (o exemplo pode ser dado com recurso a um diagrama).

- A complexidade, no pior caso, é O(n) para todas as operações. O pior caso acontece se uma sequência de inserções colocar os novos elementos sempre no mesmo lado da árvore. Exemplo:
Inserir 1, Ana -> Inserir 2, Bruno -> Inserir 3, Carla-> Inserir 4, Duarte -> Inserir 5, Eva origina uma árvore com todos os elementos à esquerda, obrigando a percorrer os n elementos da árvore, em vez de O(log n) caso a árvore esteja balanceada.



Critérios:

0.30 val. pela complexidade de cada uma das 3 operações (0.10 por cada)
0.25 val. se justifica convenientemente o pior caso
0.20 val. se fornece um exemplo

1.3 [0.75] Descreva textualmente um algoritmo utilizado para colocar os alunos presentes numa árvore binária de pesquisa, numa fila ordenada pelo número de aluno, com o aluno de número menor no início da fila e o maior no final. Deve utilizar um algoritmo recursivo.
Nota: Não é aceite código como resposta a esta questão, comentado ou não.

Critérios:

0.75 val. pela explicação correta do algoritmo
descontar 0.35 val. se justifica algoritmo correto, mas não recursivo

Resposta:
Para introduzir os alunos numa fila, de forma ordenada, pode-se utilizar uma abordagem semelhante à função imprimeOrdenado na qual percorremos a árvore, de forma recursiva, utilizando uma abordagem *inorder*:

- se existe uma subárvore à esquerda, correr o algoritmo na subárvore esquerda
- se existe aluno no nó, colocamos o aluno na fila
- se existe subárvore à direita, correr o algoritmo na subárvore direita

Critérios de correção:
- Programa desenvolvido difere significativamente das especificações e instruções do enunciado => 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.
- A componente de funcionalidade do programa é avaliada tendo como ponto de partida de casos de teste com resultado positivo. O nível de simplicidade e qualidade do código também é avaliado. Programas considerados mal estruturados, demasiado complexos, confusos ou ineficientes podem ser penalizados até 50%.
- Para a alínea 1.2 apenas são considerados o código do programa e a sua avaliação na atividade VPL (Virtual Programming Lab)
- O e-fólio só é considerado entregue com a submissão do relatório do e-fólio na plataforma moodle.
- Na atividade VPL deverá completar o código que falta, testar e gravar, confirmando, no final, que a submissão contém o código pedido.

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.

FIM