

”

Exame | Instruções para a realização de exame

ESTRUTURAS DE DADOS E ALGORITMOS FUNDAMENTAIS | 21046

Data e hora de realização

30 de junho de 2022, às 15h00, hora de Portugal Continental

Duração da prova

120 minutos + 60 minutos de tolerância (limite 18h00)

Trabalho a desenvolver

Responder às questões dos Grupos I a V.

**Leia estas informações e instruções na totalidade
antes de iniciar a resolução da prova.**

Critérios de avaliação e cotação

- As respostas às questões devem fazer sentido, ser coerentes e constituídas por palavras próprias do aluno. Não serão aceites transcrições ou traduções de livros e textos, incluindo textos de orientações de respostas de provas anteriores.
- As respostas devem ser relativamente desenvolvidas e elaboradas de modo a demonstrar o raciocínio e conhecimento que leva à resposta

final. A clareza do texto e da explicação também são levadas em conta na classificação das respostas.

– Nas questões de escrita de programas, além da explicação geral do seu funcionamento, a sua correção tem em conta critérios de proficiência e compreensibilidade do código tais como: legibilidade, indentação, estrutura e comentários.

– No caso de consulta de livros, textos, ou outros recursos, devem ser referidos na resposta os materiais consultados.

– As respostas que não respeitem as condições anteriormente mencionadas serão classificadas com zero valores ou fortemente desvalorizadas.

– As citações são indicadas por grupo e nas próprias questões.

Normas a respeitar

– Deve redigir o seu Exame no ficheiro Folha de Resolução disponibilizado na turma e preencher todos os dados do cabeçalho.

– O texto de todas as respostas deve ser introduzido pelo processador de texto, incluindo código de programas, não sendo aceites respostas escritas à mão ou por outros meios, digitalizadas e incluídas no ficheiro como imagens. São exceções figuras, diagramas e expressões matemáticas mais complicadas, desde que sejam todas de autoria do aluno, devendo ter legenda ou identificação de modo a serem referidas nos textos explicativos.

– No caso de código de programas é obrigatório a sua introdução pelo processador de texto utilizando uma fonte monoespaço (por exemplo Courier New).

– Todas as páginas do documento devem ser numeradas.

- O seu Exame não deve ultrapassar um total de 10 páginas A4, redigidas com tamanho de letra 12. O espaçamento entre linhas deve corresponder a 1,5 linhas, exceto no caso de código de programas. O formato final do ficheiro deve ser exclusivamente em formato pdf, sem restrições (destrancado). Não serão aceites outros tipos de ficheiro.
- Nomeie o ficheiro com o seu número de estudante, seguido da identificação de Exame, segundo o exemplo apresentado: 000000exame.pdf.
- O tempo de realização da prova é de 180 minutos, incluindo tolerâncias e entrega do único ficheiro pdf final com a resolução.
- Deve carregar o referido ficheiro pdf para a plataforma no dispositivo Exame até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas na composição do documento, conversão para formato pdf e submissão do ficheiro.

Votos de bom trabalho!

Paulo Pombinho

Grupo I [3 valores]

1.1. [1] Explique a diferença e entre as notações big-O, big-Omega e big-Theta.

1.2. [1] Considere a seguinte função, com complexidade $O(n)$:

```
double media (double *array, int tamanho) {...};
```

Qual a complexidade da função variância (apresentada a seguir)?

```
double variancia( double *array, int tamanho ) {  
    double soma = 0;  
    for( int i = 0; i < tamanho ; ++i ) {  
        double dif = media(array, tamanho) - array[i];  
        soma = soma + dif * dif;  
    }  
    return soma / tamanho;  
}
```

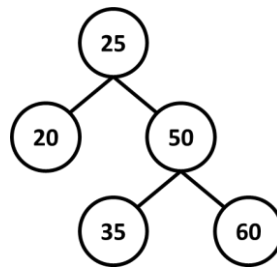
1.3. [1] Altere a função anterior de forma a torná-la mais eficiente? Indique a nova complexidade.

Grupo II [5 valores]

2.1. [1] Defina o que é uma árvore binária de pesquisa balanceada.

2.2. [1] Qual a vantagem de ter uma árvore balanceada nas operações de pesquisa? Justifique indicando a complexidade no pior caso.

2.3. [3] Considere a seguinte árvore AVL:



Insira, pela ordem indicada, os seguintes elementos, mantendo uma árvore AVL válida após cada inserção: 70, 30, 15, 32. Apresente a árvore final.

Grupo III [4 valores]

3. Considere o seguinte vetor de inteiros: 6 12 4 1 20 10. Ordene o vetor, utilizando o algoritmo indicado em cada alínea, apresentando os passos intermédios durante a ordenação.

3.1. [2] Selection Sort:

3.2. [2] Merge Sort:

Grupo IV [4 valores]

4. Considere uma estrutura de dados do tipo pilha de inteiros positivos, utilizando uma implementação com listas ligadas. Com a seguinte definição da classe:

```
class NoPilha {
public:
    int elemento;
    NoPilha* proximo;
};
```

- 4.1. [1] Defina um método que devolve um novo nó, inicializado com elemento, mas sem apontar para outros nós:

```
NoPilha* novoNo(int elemento);
```

- 4.2. [1] Defina o método push:

```
void push(noPilha** raiz, int elemento);
```

- 4.3. [2] Defina o método pop, retorne -1 caso a pilha esteja vazia:

```
int pop(noPilha** raiz);
```

Grupo V [4 valores]

5. Considere uma implementação de uma árvore binária com elementos inteiros, utilizando a seguinte classe para os seus nós:

- 5.1. [1] Defina o método que permite saber a altura de uma árvore. Implemente o método de forma recursiva:

```
int altura(no* raiz);
```

- 5.2. [3] Defina o método que verifica se a árvore está balanceada (pode utilizar o método definido na alínea anterior). Implemente o método de forma recursiva.

```
bool estaBalanceada(no* raiz);
```

FIM