

”

Exame | Instruções para a realização de exame

ESTRUTURAS DE DADOS E ALGORITMOS FUNDAMENTAIS | 21046

Data e hora de realização

20 de setembro 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 monoespacia (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. Indique, justificando, qual a complexidade das funções seguintes:

1.1. [1] Função que devolve o índice da primeira ocorrência de um número:

```
int encontraPrimeiroIndice (numero, vetor) {  
  
    for (i = 0; i < vetor.length; i++)  
        if (vetor[i] == numero)  
            return i;  
  
    return -1;  
  
}
```

1.2. [1] Função que devolve a soma de um vetor, sequencial e ordenado (vetor = [1, 2, 3, 4, 5...]):

```
int somaVetorSequencial(vetor) {  
  
    return vetor.length * (vetor.length + 1) / 2;  
  
}
```

1.3. [1] Função que encontra o fatorial de um número:

```
int fatorial(num) {  
  
    if (num == 0 || num == 1)  
        return 1;  
  
    return n * fatorial(num - 1);  
  
}
```

Grupo II [5 valores]

- 2.1. [1] Explique a diferença, numa árvore binária, entre navegação em largura e navegação em profundidade.
- 2.2. [1] Indique, justificando, quais as vantagens de utilizar uma lista ligada em alternativa a um *array*.
- 2.3. [3] Considere uma tabela de dispersão com dimensão 13 e função de hash $h(x) = x\%13$ e em que as colisões são resolvidas com sondagem quadrática. Considere a tabela vazia e a inserção dos elementos 4, 6, 12, 63, 14, 1, 27 pela ordem indicada. Indique a tabela final, justificando os passos intermédios.

Grupo III [4 valores]

3. Considere o seguinte vetor de inteiros: 4 6 2 12 5 15. Ordene o vetor, utilizando o algoritmo indicado em cada alínea, apresentando e explicando os passos intermédios durante a ordenação.

3.1. [2] Insert Sort:

3.2. [2] Quick Sort:

Grupo IV [4 valores]

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

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

- 4.1. [2] Defina um método que insere um novo nó na fila, com o elemento passado por parâmetro:

```
void insere(int elemento);
```

- 4.2. [2] Defina um método que remove um elemento da fila:

Grupo V [4 valores]

5. Considere uma implementação de um *min heap* com elementos inteiros, em que a raiz contém o menor elemento, utilizando a seguinte classe:

```
class MinHeap {
    int* vetor_heap;
    int tamanho_max;
    int tamanho;
};
```

Considere ainda que dispõem da função `troca(int *x, int *y)` que troca dois inteiros.

- 5.1. [2] Defina o método que permite inserir um novo elemento na estrutura:

```
void inserir(int elem);
```

- 5.2. [2] Explique como implementaria o método que remove, e devolve, o menor elemento da estrutura.
(Pode utilizar pseudo-código, não precisa escrever código).

```
int retiraMenorElemento();
```

FIM