



Exame | Instruções para a realização de exame



Estruturas de Dados e Algoritmos Fundamentais | 21046

Data de Realização

Dia 18 de setembro de 2023 – Versão 1

Duração da prova

120 minutos, mais 30 minutos de tolerância

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 cotações são indicadas por grupo e nas próprias questões.
- 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 que não respeitem estas condições serão classificadas com zero valores ou fortemente desvalorizadas.
- 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. À simples indicação do resultado é atribuída a cotação zero.
- Nas questões de escrita de programas, a sua correção tem em conta critérios de proficiência e compreensibilidade do código tais como: legibilidade, indentação, estrutura, comentários e explicação geral do seu funcionamento.

Normas a respeitar

- Se o seu exemplar não estiver completo ou nele se verificar qualquer outra deficiência, por favor dirija-se ao professor vigilante.
- A prova deve ser resolvida na sua totalidade em folhas de respostas.
- Nas respostas, tenha a preocupação de utilizar uma letra legível.
- Todas as respostas devem ser escritas unicamente com caneta azul ou preta.
- A prova é SEM CONSULTA. Todos os elementos necessários à resolução são fornecidos no enunciado.

Votos de bom trabalho!

Grupo I [3 valores]

- 1.1. [1] Utilizando a definição, prove que $f(n) = n^2 + 2n + 10$ é $\Omega(n^2)$.
- 1.2. Para cada um dos seguintes pares de funções $f(n)$ e $g(n)$, indique se $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, $f(n) = \Theta(g(n))$ ou nenhum dos casos. Justifique com base apenas na ordem de grandeza relativa das funções.
- 1.2.1. [0.5] $f(n) = 1 + \sqrt{n}$, $g(n) = 10 + 2 \log(n)$
- 1.2.2. [0.5] $f(n) = 2^n + 100$, $g(n) = 2^{n+1} + 500$
- 1.3. [1] Considere a complexidade do seguinte segmento de código em termos do nº $f(n)$ de operações aritméticas realizadas na variável a . Determine justificando a expressão de $f(n)$ e indique a sua complexidade na notação $O(\cdot)$.

```
for(a=0,i=2; i<=n*n; i*=2)
    for(j=i; j<=n; ++j)
        a++;
```

Grupo II [5 valores]

- 2.1. Considere uma árvore de promoção (Splay Tree) inicialmente vazia.
- 2.1.1. [1] Insira na árvore as chaves 12, 4, 13, 15, 3, 5, 11 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Nas alíneas seguintes considere a árvore obtida como a árvore "original".
- 2.1.2. [1] Remova da árvore original a chave 12 utilizando o algoritmo de remoção por cópia (Deletion by Copying) com a chave antecessora. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

2.1.3. [1] Considere que na árvore original foi efetuado um acesso à chave 5. Desenhe a árvore obtida após o acesso (total de 1 desenho) justificando os passos intermédios / raciocínio.

2.2. [2] Construa uma função recursiva que dado um vetor v com n elementos inteiros calcule a soma máxima dos seus elementos pares consecutivos. Por exemplo, para o vetor com os elementos 1, 4, 2, 7, 3, 6, 3, 2, 8, 1 deve retornar 10.

Grupo III [4 valores]

3.1. [2] Considere uma tabela T de dispersão (hash) com dimensão 11 e função de hash $h(x) = x \bmod 11$. As colisões são resolvidas com sondagem (probing) quadrática. Considerando a tabela inicialmente vazia, determine o conteúdo final da tabela após a inserção das chaves 11, 5, 2, 17, 16, 27, 4 pela ordem apresentada. Justifique os cálculos efetuados para cada inserção.

3.2. [2] Considere o vetor [3 7 8 9 5 1 4 2 6]. Ordene o vetor, utilizando o algoritmo de ordenação Quick Sort, apresentando e justificando os passos intermédios durante a ordenação.

Grupo IV [4 valores]

4. Pretende-se conceber em C++ uma estrutura de dados tipo lista simplesmente ligada (single linked list) em que os itens são inteiros. A lista deve suportar as operações de: (i) Inserir um item no início da lista (método `insert_0`); (ii) Remover e retornar o item numa posição da lista (método `delete_pos`).

4.1. [1.5] Implemente o método "insert_0".

4.2. [2.5] Implemente o método "delete_pos".

Grupo V [4 valores]

5. Pretende-se conceber em C++ uma estrutura de dados tipo árvore de pesquisa binária (BST - Binary Search Tree) em que os itens são genéricos. A BST deve suportar a operação de remover um item da árvore utilizando o algoritmo de remoção por cópia simétrica (método `deleteByCopyingSim`).

5.1. [4] Implemente o método "deleteByCopyingSim". O método deve retornar 0 em caso de sucesso e -1 se o item indicado não for encontrado.

FIM