



Exame | Instruções para a realização de exame



Estruturas de Dados e Algoritmos Fundamentais | 21046

Data de Realização

Dia 22 de setembro de 2021

Data e Hora de Limite de Entrega

Dia 22 de setembro de 2021, hora de acordo com o especificado pela plataforma.

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 cotações 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!

Grupo I [3 valores]

- 1.1. [1] Utilizando a definição, prove que $f(n) = n^3 + 10n^2 + 200$ é $O(n^3)$.
- 1.2. Para cada um dos seguintes pares de funções $f(n)$ e $g(n)$, indique (não necessita justificação) se $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, $f(n) = \Theta(g(n))$ ou nenhum dos casos.
- 1.2.1. [0.5] $f(n) = 20 \log_2 n^2$, $g(n) = \sqrt{n}$
- 1.2.2. [0.5] $f(n) = \frac{10}{n} + 2^n$, $g(n) = n^3 + 100$
- 1.3. [1] Considere a complexidade do seguinte segmento de código em termos do n^o $f(n)$ 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=1; j<=n; ++j)
    a++;
```

Grupo II [5 valores]

2. Considere uma árvore binária de promoção (Splay Tree) inicialmente vazia.
- 2.1. [1] Insira na árvore as chaves 6, 3, 4, 8, 2, 1, 9, 7 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Justifique a inserção da chave 7. Nas alíneas seguintes considere a árvore obtida como a árvore "original".
- 2.2. [2] Considere que na árvore original foram efetuados acessos às chaves 8, 3 e 7 pela ordem indicada. Desenhe a árvore obtida após cada acesso (total de 3 desenhos). Justifique os passos intermédios / raciocínio.
- 2.3. [2] Remova da árvore original a chave 6 utilizando o algoritmo de remoção por fusão (Deletion by Merging), escolhendo a opção em que o nó removido é substituído pelo seu filho direito. Desenhe a árvore obtida. Justifique os passos intermédios / raciocínio.

Grupo III [4 valores]

- 3.1. [2] Considere uma tabela T de dispersão (hash) com dimensão 9 e função de hash $h(x) = x \bmod 9$. As colisões são resolvidas com sondagem (probing) linear. Considerando a tabela inicialmente vazia, determine o conteúdo final da tabela após a inserção das chaves 5, 6, 14, 15, 9, 19, 0 pela ordem apresentada. Justifique os cálculos efetuados para cada inserção.
- 3.2. [2] Considere o vetor [7 1 8 5 4 6 3 9 2]. Ordene o vetor utilizando uma versão simplificada do algoritmo QuickSort (sem questões de eficiência na troca de elementos para a divisão dos vetores). Utilize para pivot o elemento do meio. Explique de um modo geral o funcionamento do algoritmo e indique justificando a sequência de vetores/passos correspondentes às iterações principais do algoritmo.

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 do fim da lista (método `delete_end`).

Na resolução das alíneas seguintes pode criar métodos e construtores adicionais que achar convenientes. Explique em termos gerais o funcionamento do código e indique explicitamente situações especiais ou casos particulares considerados.

- 4.1. [1] Defina as classes que entender necessárias para a implementação da lista. Defina apenas atributos e métodos, não inclua código para os métodos. Justifique cada atributo que incluir nas classes.
- 4.2. [1] Implemente o método "insert_0".
- 4.3. [2] Implemente o método "delete_end".

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 as operações de: (i) Inserir um item na árvore (método `insert`); (ii) Remover um item da árvore utilizando o algoritmo de remoção por fusão (método `deleteByMerging`).

Na resolução das alíneas seguintes pode criar métodos e construtores adicionais que achar convenientes. Explique em termos gerais o funcionamento do código e indique explicitamente situações especiais ou casos particulares considerados.

- 5.1. [1] Defina as classes que entender necessárias para a implementação da BST. Defina apenas atributos e métodos, não inclua código para os métodos. Justifique cada atributo que incluir nas classes.
- 5.2. [1] Implemente o método "insert".
- 5.3. [2] Implemente o método "deleteByMerging". O método deve retornar 0 em caso de sucesso e -1 se o item indicado não for encontrado.

FIM