



Exame | Instruções para a realização de exame



Estruturas de Dados e Algoritmos Fundamentais | 21046

Data de Realização

Dia 1 de setembro de 2020

Data e Hora de Limite de Entrega

Dia 1 de setembro de 2020, 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 (Exemplo, sujeito a alterações)

- 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 (Exemplo, sujeito a alterações)

- 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 XXX 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 $f(n) = n + \sqrt{n}$ é $O(n^2)$.
- 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.
- 1.2.1. [0.5] $f(n) = 10n \log_2(n)$, $g(n) = n\sqrt{n}$
- 1.2.2. [0.5] $f(n) = 1000 + 2^n$, $g(n) = n^2$
- 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=1; i<=n; i++)
  for(j=i; j<=n; j++)
    for(k=1; k<=n; k++)
      a++;
```

Grupo II [5 valores]

- 2.1. [2] Considere uma árvore binária do tipo max Heap inicialmente vazia na qual foram inseridas as chaves 7, 3, 2, 1, 5, 6, 9, 8 pela ordem indicada. Indique na forma de vetor o Heap após cada inserção (apresente 8 vetores). Justifique os passos intermédios / raciocínio.
- 2.2. [1] Remova do Heap obtido na alínea anterior a chave de maior valor e indique na forma de vetor o Heap resultante. Justifique os passos intermédios / raciocínio.
- 2.3. [2] Construa uma função recursiva que dado um vetor v com n elementos determine o número de elementos que são múltiplos de 3. O protótipo da função é dado por `int cmult3(int v[], int n);`. Caracterize o tipo de recursão que utilizou na função.

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) quadrática. Considerando a tabela inicialmente vazia, determine o conteúdo final da tabela após a inserção das chaves 12, 14, 3, 7, 11, 21, 2 pela ordem apresentada. Justifique os cálculos efetuados para cada inserção.
- 3.2. [2] Considere o vetor [9 4 3 6 2 5 8 7 1]. Indique a sequência de passos para a sua ordenação 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. Justifique de um modo geral o funcionamento do algoritmo.

Grupo IV [4 valores]

4. Pretende-se conceber em C++ uma estrutura de dados tipo pilha (Stack) implementada por uma lista simplesmente ligada (Single Linked List) em que os itens são inteiros. A pilha deve suportar as operações de: (i) Inserir um item (método push); (ii) Remover e retornar um item (método pop).

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 pilha. Defina apenas atributos e métodos, não inclua código para os métodos. Justifique cada atributo que incluir nas classes.
- 4.2. [1.5] Implemente o método "push".
- 4.3. [1.5] Implemente o método "pop". Admita que a pilha não está vazia.

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) Determinar e retornar a altura da árvore (método height).

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. [0.5] Apresente a definição de altura (height) de uma árvore. Indique a altura para os casos particulares de uma árvore vazia e uma árvore só com um nó.
- 5.4. [1.5] Implemente o método "height".

FIM