E-fólio Global | Instruções para a realização do E-fólio

Estruturas de Dados e Algoritmos Fundamentais | 21046

Data de Realização

Dia 17 de setembro de 2024

Duração da prova

90 minutos, mais 60 minutos de tolerância

Trabalho a desenvolver

Responder às questões dos Grupos I a III.

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.
- Exceto indicação contrária, todas 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.
- 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.
- No caso de consulta de livros, textos, ou outros recursos, devem ser referidos na resposta os materiais consultados.

Instruções e Normas a respeitar

- Este documento tem 4 enunciados: E0, E1, E2 e E3. Calcule o resto R da divisão inteira do seu nº de estudante por 4 e resolva o respetivo enunciado ER. Apenas é necessário considerar os últimos 2 dígitos (dezenas e unidades). Exemplo: Se o seu nº de estudante é -----38, então o resto inteiro de 38/4=9 é R=2 e é-lhe atribuído o enunciado E2. A resolução de outro enunciado é classificada com 0 valores.

- Deve redigir o seu e-fólio no ficheiro Folha de Resolução disponibilizado na turma e preencher todos os dados do cabeçalho, incluindo a indicação do enunciado a resolver que lhe é atribuído, em função do nº de estudante.

- 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 e-fólio deve ser constituído por 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 do e-fólio, seguido do enunciado que lhe foi atribuido, segundo o exemplo apresentado: 000000efolioGlobalE0.pdf.

- Deve carregar o referido ficheiro pdf para a plataforma no dispositivo E-fólio Global 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.

- O ficheiro a entregar não deve exceder 8 MB.

Votos de bom trabalho!

Enunciado E0

Grupo I [3 valores]

1.1. [1] Utilizando a definição, prove que $f(n) = n^2 + n^{3/2} + 10$ 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. Justifique a sua resposta com base apenas na ordem de grandeza relativa das funções.

1.2.1. [0.5] $f(n) = n \log n + 1, \;\; g(n) = \log n^2 + 100$

1.2.2. [0.5] $f(n) = 3^{n+10} + n^2 + 2, \;\; g(n) = 3^{n+2} + n^3 + 2$

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 $O(.)$.

```
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. Considere uma árvore de pesquisa binária (BST) inicialmente vazia.

2.1.1. [1] Insira na árvore as chaves 10, 5, 3, 15, 8, 17, 7, 9, 12 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Justifique os passos intermédios / raciocínio apenas para as duas últimas inserções. Nas alíneas seguintes considere a árvore obtida como a árvore "original".

2.1.2. [1] Remova da árvore original a chave 10 utilizando o algoritmo de remoção por fusão (Deletion by Merging). Das várias opções possíveis escolha a que resulta numa árvore em que a subárvore direita da raiz tem maior altura. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

2.1.3. [1] Efetue na árvore original uma rotação de 5 em torno de 10. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

2.2. [2] Considere uma árvore B (B-Tree) de ordem 3 inicialmente contendo apenas o nó raiz com as chaves 0 e 1. Desenhe a árvore após cada uma das seguintes operações de inserção (I) e remoção (R) pela ordem indicada: I 2, 3 ,4 ,5 ,6; R 5; (total de 6 desenhos). Justifique os passos intermédios / raciocínio para cada operação.

Enunciado E0

Grupo III [4 valores]

3. Considere o vetor [5 6 4 1 3 9 2 7 8]. Ordene o vetor utilizando o algoritmo de ordenação indicado, apresentando e justificando a sequência de vetores obtida correspondente às iterações principais do algoritmo.

3.1. [2] Algoritmo de ordenação Shell Sort, utilizando os incrementos 1, 3, 4.

3.2. [2] Algoritmo de ordenação por fusão (Merge Sort).

Enunciado E1

Grupo I [5 valores]

1.1. Considere uma árvore de pesquisa binária (BST) inicialmente vazia.

1.1.1. [1] Insira na árvore as chaves 10, 5, 3, 15, 8, 17, 7, 9, 12 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Justifique os passos intermédios / raciocínio apenas para as duas últimas inserções. Nas alíneas seguintes considere a árvore obtida como a árvore "original".

1.1.2. [1] Remova da árvore original a chave 10 utilizando o algoritmo de remoção por fusão (Deletion by Merging). Das várias opções possíveis escolha a que resulta numa árvore em que a subárvore direita da raiz tem maior altura. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

1.1.3. [1] Efetue na árvore original uma rotação de 5 em torno de 10. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

1.2. [2] Considere uma árvore B (B-Tree) de ordem 3 inicialmente contendo apenas o nó raiz com as chaves 0 e 1. Desenhe a árvore após cada uma das seguintes operações de inserção (I) e remoção (R) pela ordem indicada: I 2, 3 ,4 ,5 ,6; R 5; (total de 6 desenhos). Justifique os passos intermédios / raciocínio para cada operação.

Grupo II [4 valores]

2. Considere o vetor [9 2 3 6 5 4 8 1 7]. Ordene o vetor utilizando o algoritmo de ordenação indicado, apresentando e justificando a sequência de vetores obtida correspondente às iterações principais do algoritmo.

2.1. [2] Algoritmo de ordenação Shell Sort, utilizando os incrementos 1, 3, 4.

2.2. [2] Algoritmo de ordenação por fusão (Merge Sort).

Enunciado E1

Grupo III [3 valores]

3.1. [1] Utilizando a definição, prove que $f(n) = n^2 + n^{3/2} + 10$ é $\Omega(n^2)$.

3.2. 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 a sua resposta com base apenas na ordem de grandeza relativa das funções.

3.2.1. [0.5] $f(n) = n + n^3 + 2$, $g(n) = \log n^4 + 100$

3.2.2. [0.5] $f(n) = 1.5^n + n^2 + 2$, $g(n) = n^2 \log n + 1$

3.3. [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 $O(.)$.

```
for(a=0,i=1; i<=n; ++i)
   for(j=1; j<=n; ++j)
      for(k=j; k<=n; ++k)
         a++;
```

Enunciado E2

Grupo I [4 valores]

1. Considere o vetor [5 6 4 1 3 9 2 7 8]. Ordene o vetor utilizando o algoritmo de ordenação indicado, apresentando e justificando a sequência de vetores obtida correspondente às iterações principais do algoritmo.

1.1. [2] Algoritmo de ordenação Shell Sort, utilizando os incrementos 1, 3, 4.

1.2. [2] Algoritmo de ordenação por fusão (Merge Sort).

Grupo II [5 valores]

2.1. Considere uma árvore de pesquisa binária (BST) inicialmente vazia.

2.1.1. [1] Insira na árvore as chaves 14, 4, 3, 18, 11, 19, 7, 13, 16 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Justifique os passos intermédios / raciocínio apenas para as duas últimas inserções. Nas alíneas seguintes considere a árvore obtida como a árvore "original".

2.1.2. [1] Remova da árvore original a chave 14 utilizando o algoritmo de remoção por fusão (Deletion by Merging). Das várias opções possíveis escolha a que resulta numa árvore em que a subárvore esquerda da raiz tem maior altura. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

2.1.3. [1] Efetue na árvore original uma rotação de 18 em torno de 14. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

2.2. [2] Considere uma árvore B (B-Tree) de ordem 3 inicialmente contendo apenas o nó raiz com as chaves 5 e 6. Desenhe a árvore após cada uma das seguintes operações de inserção (I) e remoção (R) pela ordem indicada: I 4, 3, 2, 1, 0; R 1; (total de 6 desenhos). Justifique os passos intermédios / raciocínio para cada operação.

Enunciado E2

Grupo III [3 valores]

3.1. [1] Utilizando a definição, prove que $f(n) = n^2 + n^{3/2} + 10$ é $\Omega(n^2)$.

3.2. 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 a sua resposta com base apenas na ordem de grandeza relativa das funções.

3.2.1. [0.5] $f(n) = n + n^3 + 2$, $g(n) = \log n^4 + 100$

3.2.2. [0.5] $f(n) = 1.5^n + n^2 + 2$, $g(n) = n^2 \log n + 1$

3.3. [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 $O(.)$.

```
for(a=0,i=1; i<=n; ++i)
   for(j=1; j<=n; ++j)
      for(k=j; k<=n; ++k)
         a++;
```

Enunciado E3

Grupo I [4 valores]

1. Considere o vetor [9 2 3 6 5 4 8 1 7]. Ordene o vetor utilizando o algoritmo de ordenação indicado, apresentando e justificando a sequência de vetores obtida correspondente às iterações principais do algoritmo.

1.1. [2] Algoritmo de ordenação Shell Sort, utilizando os incrementos 1, 3, 4.

1.2. [2] Algoritmo de ordenação por fusão (Merge Sort).

Grupo II [3 valores]

2.1. [1] Utilizando a definição, prove que $f(n) = n^2 + n^{3/2} + 10$ é $O(n^2)$.

2.2. 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 a sua resposta com base apenas na ordem de grandeza relativa das funções.

2.2.1. [0.5] $f(n) = n \log n + 1, \;\; g(n) = \log n^2 + 100$

2.2.2. [0.5] $f(n) = 3^{n+10} + n^2 + 2, \;\; g(n) = 3^{n+2} + n^3 + 2$

2.3. [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 $O(.)$.

```
for(a=0,i=1; i<=n; ++i)
   for(j=i; j<=n; ++j)
      for(k=1; k<=n; ++k)
         a++;
```

Enunciado E3

Grupo III [5 valores]

3.1. Considere uma árvore de pesquisa binária (BST) inicialmente vazia.

3.1.1. [1] Insira na árvore as chaves 14, 4, 3, 18, 11, 19, 7, 13, 16 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Justifique os passos intermédios / raciocínio apenas para as duas últimas inserções. Nas alíneas seguintes considere a árvore obtida como a árvore "original".

3.1.2. [1] Remova da árvore original a chave 14 utilizando o algoritmo de remoção por fusão (Deletion by Merging). Das várias opções possíveis escolha a que resulta numa árvore em que a subárvore esquerda da raiz tem maior altura. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

3.1.3. [1] Efetue na árvore original uma rotação de 18 em torno de 14. Desenhe a árvore obtida justificando os passos intermédios / raciocínio.

3.2. [2] Considere uma árvore B (B-Tree) de ordem 3 inicialmente contendo apenas o nó raiz com as chaves 5 e 6. Desenhe a árvore após cada uma das seguintes operações de inserção (I) e remoção (R) pela ordem indicada: I 4, 3, 2, 1, 0; R 1; (total de 6 desenhos). Justifique os passos intermédios / raciocínio para cada operação.

FIM