

A prova é individual. Cada estudante deve enviar suas respostas. Recomenda-se consultar principalmente o livro de CLRS, usado como livro-texto na disciplina. Mas outras fontes também podem ser necessárias. Para cada questão objetiva, a correção levará em conta exclusivamente a marcação de só uma resposta correta entre as cinco alternativas apresentadas em cada questão. A correta organização das folhas de respostas é responsabilidade do aluno. A interpretação faz parte da prova. O nome do(a) estudante é item obrigatório no arquivo com as respostas. O arquivo com as respostas deve ser enviado exclusivamente na atividade criada para essa finalidade na página da disciplina no Campus Virtual. Não serão aceitas provas enviadas com atraso. Há tempo bastante suficiente para fazer e enviar a prova. Não serão aceitas provas após o prazo por qualquer motivo, como atraso no envio das respostas motivado por eventuais falhas de conexões com a internet, falta de energia elétrica, congestionamento das linhas de comunicação, bem como outros fatores de ordem técnica que impossibilitem a conexão ou a transferência de dados. Essas eventualidades não serão aceitas como argumento para envio das respostas após o prazo. Recomenda-se que as respostas sejam encaminhadas com prudente antecedência. Cada uma das questões 4, 5, 7 e 9 valem 9 pontos. São 8 pontos para cada uma das demais questões.

Nome: **Pedro Antônio de Souza (201810557)**

1 Em um algoritmo para o problema do caixeiro viajante, a partir de cada vértice v , o próximo vértice a ser percorrido é determinado pela aresta de menor custo. Esse algoritmo foi projetado pela técnica:

a) Programação dinâmica b) Branch and bound c) Busca exaustiva d) Divisão e conquista ☒ e) Algoritmos gulosos

2 A técnica utilizada para projetar um algoritmo que divide um problema em subproblemas menores e resolve independentemente cada subproblema é:

a) Programação dinâmica b) Branch and bound c) Busca exaustiva ☒ d) Divisão e conquista e) Algoritmos gulosos

3 A técnica a ser utilizada para se desenvolver um algoritmo que forneça a solução para um problema de decisão é:

a) Programação dinâmica ☒ b) Bracktracking c) Busca exaustiva d) Divisão e conquista e) Algoritmos gulosos

4 Identifique se cada sentença é verdadeira ou falsa em cada questão e escolha a alternativa correta.

I) Se subproblemas podem ser aninhados recursivamente dentro de problemas maiores, então, a programação dinâmica pode ser utilizada. Nesses casos, há uma relação direta entre os valores dos subproblemas em determinada sequência de decisões que leva ao valor de um problema maior que tais subproblemas.

II) Programação dinâmica é uma técnica para propósito geral, baseada na decomposição do problema, aplicada tipicamente a problemas de otimização. O termo programação refere-se ao código computacional e não a planejamento.

III) A cada estágio de um algoritmo projetado por programação dinâmica, as decisões são baseadas em todas as decisões realizadas em estágios anteriores e pode reconsiderar o caminho de estágios anteriores.

IV) A programação dinâmica é útil quando um algoritmo por outra técnica apresenta complexidade exponencial. Quando a programação dinâmica pode ser aplicada, geralmente, o algoritmo tem tempo de execução assintótico menor que um algoritmo construído por força bruta, por exemplo. Apesar disso, a técnica sempre fornece soluções aproximadas à solução ótima de um problema de otimização.

V) Na programação dinâmica, busca-se resolver um problema complexo ao decompô-lo em subproblemas menores. Um algoritmo construído por programação dinâmica é aplicável quando os subproblemas são dependentes, ou seja, quando subproblemas compartilham subproblemas.

a) I: verdadeira; II: verdadeira; III: falsa; IV: verdadeira; V: falsa.

☒ b) I: verdadeira; II: falsa; III: verdadeira; IV: falsa; V: verdadeira.

c) I: falsa; II: verdadeira; III: falsa; IV: falsa; V: falsa.

d) Todas são falsas.

e) Todas são verdadeiras.

5 Atribua verdadeiro ou falso a cada sentença e assinale a alternativa correta.

- I) A similaridade entre algoritmos gulosos e programação dinâmica é que algoritmos projetados por ambas as técnicas pré-calculam soluções ótimas para subproblemas.
- II) A similaridade entre divisão e conquista e programação dinâmica (com tabelamento) é que ambas utilizam uma abordagem *top-down*.
- III) A similaridade entre algoritmos gulosos e divisão e conquista é que algoritmos projetados por ambas as técnicas fazem as melhores escolhas em cada iteração.
- IV) Algoritmo por programação dinâmica e por *branch em bound* sempre retornam a solução ótima do problema de otimização.
- V) Um algoritmo por força bruta é um algoritmo ingênuo. Logo, sempre haverá um algoritmo melhor que um algoritmo por força bruta para todos os problemas computacionais.

- a) I: falsa; II: falsa; III: verdadeira; IV: falsa; V: verdadeira.
- ☒ b) I: verdadeira; II: verdadeira; III: falsa; IV: verdadeira; V: falsa.
- c) I: falsa; II: falsa; III: falsa; IV: verdadeira; V: falsa.
- d) Todas são verdadeiras.
- e) Todas são falsas.

6 Atribua verdadeiro ou falso a cada sentença e assinale a alternativa correta.

- I) A programação dinâmica pode ser utilizada quando é necessária a solução ótima do problema.
- II) A programação dinâmica pode ser utilizada quando a solução tem subestrutura ótima.
- III) Um algoritmo por programação dinâmica para o problema em contexto sempre será mais rápido que um algoritmo guloso.
- IV) Um algoritmo por programação dinâmica para o problema em contexto sempre será mais rápido que um algoritmo por divisão e conquista.
- V) A programação dinâmica é utilizada quando se deseja um algoritmo mais rápido que um algoritmo por força bruta.

- a) I: falsa; II: falsa; III: verdadeira; IV: falsa; V: verdadeira.
- ☒ b) I: verdadeira; II: verdadeira; III: falsa; IV: falsa; V: verdadeira.
- c) I: falsa; II: falsa; III: falsa; IV: verdadeira; V: falsa.
- d) Todas são verdadeiras.
- e) Todas são falsas.

7 Considere o seguinte problema: dado um conjunto de inteiros não negativos e um valor k , é preciso determinar se há um subconjunto do dado conjunto em que a soma dos números seja k . Entre as técnicas força bruta, divisão e conquista, programação dinâmica, algoritmos gulosos e *branch and bound*, a técnica a ser utilizada para encontrar a solução ótima em menor tempo para o problema é: Programação dinâmica

8 Responda as técnicas que foram utilizadas para se projetar os algoritmos:

Bellman-Ford: Programação dinâmica
Busca linear: Força bruta
Floyd-Warshall: Programação dinâmica
Kruskal: Algoritmo guloso
Prim: Algoritmo guloso

9 Determine o número de dígitos comparados (com e sem sucesso) pelo algoritmo por força bruta na busca por 001 na *string* binária composta de 100 zeros. Serão comparados 294 dígitos.

10 Considere um algoritmo projetado por divisão e conquista com tempo de execução expresso pela equação de recorrência $t(n) = 256t\left(\frac{n}{4}\right) + q \cdot n^r$ se $n > 1$, em que n é o tamanho da entrada do algoritmo. Dado que $t(1) = \Theta(1)$ e $q, r \in \mathbb{R}^+$, a complexidade do algoritmo é.

- a) $\Theta(n^4)$, se $r \leq 4$; e $\Theta(n^r)$, se $r > 4$ e existe $c \in \mathbb{R}^+$, tal que $\frac{256}{4^r} \leq c < 1$.
- b) $\Theta(n^4)$, se $r < 4$; e $\Theta(n^r)$, se $r \geq 4$ e existe $c \in \mathbb{R}^+$, tal que $\frac{256}{4^r} \leq c < 1$.
- c) $\Theta(n^r)$, se $r \leq 4$; e $\Theta(n^4)$, se $r > 4$ e existe $c \in \mathbb{R}^+$, tal que $\frac{256}{4^r} \leq c < 1$.
- d) $O(n^4 \cdot \lg n)$ se $r \geq 4$.
- ☒ e) Nenhuma das quatro alternativas acima está correta.

11 Sejam as equações de recorrência $t_a(n) = 624t_a\left(\frac{n}{5}\right) + n^4$, $t_b(n) = 27t_b\left(\frac{n}{3}\right) + n^3$ e $t_c(n) = 24t_c\left(\frac{n}{5}\right) + n^2$. Considere $t_a \in \Theta(n^{k_a})$, $t_b \in \Theta(f(n))$ e $t_c \in \Theta(n^{k_c})$. Ao calcular $g(n)$, pode-se afirmar que:

- ☒ a) $(\exists \epsilon \in \mathbb{R}^+) \Theta(n^{k_a}) = \Theta(n^{k_c+\epsilon})$ e $\Theta(f(n))$ é assintoticamente menor que $\Theta(n^{k_a})$ e assintoticamente maior que $\Theta(n^{k_c})$.
- b) $(\exists \epsilon \in \mathbb{R}^+) \Theta(n^{k_c}) = \Theta(n^{k_a+\epsilon})$ e $\Theta(f(n))$ é assintoticamente maior que $\Theta(n^{k_a})$ e assintoticamente menor que $\Theta(n^{k_c})$.
- c) $(\exists \epsilon \in \mathbb{R}^+) \Theta(n^{k_a}) = \Theta(n^{k_c+\epsilon})$ e $\Theta(f(n))$ é assintoticamente maior que $\Theta(n^{k_a})$ e assintoticamente menor que $\Theta(n^{k_c})$.
- d) $(\exists \epsilon \in \mathbb{R}^+) \Theta(n^{k_c}) = \Theta(n^{k_a+\epsilon})$ e $\Theta(f(n))$ é assintoticamente menor que $\Theta(n^{k_a})$ e assintoticamente maior que $\Theta(n^{k_c})$.
- e) Nenhuma das quatro alternativas acima está correta.

12 Atribua verdadeiro ou falso a cada sentença e assinale a alternativa correta.

I) A teoria sobre \mathcal{NP} -Completeness é aplicada a problemas de decisão. Problemas de decisão têm duas respostas possíveis: sim ou não.

II) Em um problema de decisão, também tem-se que a solução é dada ao se minimizar ou maximizar uma função objetivo do problema.

III) A teoria sobre \mathcal{NP} -Completeness é restrita a problemas de decisão. Logo, não se pode estender suas implicações para problemas de otimização.

IV) Se a função objetivo de um problema de otimização é relativamente fácil de ser avaliada, o problema de decisão correspondente pode ser mais fácil que o problema de otimização.

V) Se fosse possível resolver deterministicamente o problema do caixeiro viajante em tempo polinomial, então, seria possível resolver deterministicamente o problema de decisão associado em tempo polinomial.

a) I: verdadeira; II: verdadeira; III: verdadeira; IV: falsa; V: falsa.

b) I: falsa; II: falsa; III: verdadeira; IV: verdadeira; V: falsa.

c) I: falsa; II: verdadeira; III: falsa; IV: falsa; V: verdadeira.

☒ d) I: verdadeira; II: falsa; III: falsa; IV: verdadeira; V: verdadeira.

e) Todas são falsas.