

Nome: _____

Nota: _____

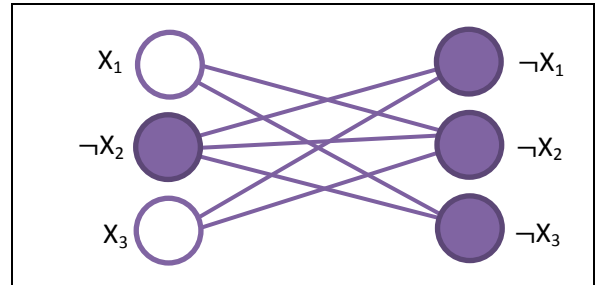
Questão 1 _(4,8) Verifique as afirmações abaixo e responda se elas são verdadeiras ou falsas. Para as falsas, justifique a sua resposta:

- a) Os algoritmos de programação dinâmica geralmente possuem complexidades de tempo e espaço melhores (menores) do que algoritmos gulosos (ou seja, são mais eficientes).
- b) Os algoritmos de *backtracking* baseados em uma busca em profundidade costumam ser mais rápidos e ocupam mais espaço na memória quando comparados aos algoritmos de *backtracking* baseados em uma busca em largura.
- c) Se um problema NP-Completo qualquer for resolvido em tempo polinomial por um algoritmo qualquer, seja ele determinista ou não determinista, então podemos afirmar que $P = NP$.
- d) Se $NP\text{-Completo} \cap P \neq \emptyset$ então $NP\text{-Completo} \subseteq P$ e logo $P = NP$.
- e) Um algoritmo é dito que executa em tempo pseudo-polinomial quando a sua complexidade é expressa por um polinômio de grau muito alto e, por mais que sua complexidade seja expressa por um polinômio, ele demora muito tempo para computar uma solução.
- f) Para provar que um dado problema Y é NP-Completo, deve-se reduzir um problema NP-Completo X em Y em tempo polinomial ($X \leq_p Y$) e mostrar que Y não pertence à classe de problemas P usando a ideia de certificado, por exemplo.
- g) Se for descoberto que $P = NP$ então todos os problemas considerados NP-Difícil (ou *NP-Hard*) podem ser resolvidos em tempo polinomial por um algoritmo determinista, com exceção dos problemas indecidíveis.
- h) Se um problema é dito que pode ser resolvido em tempo polinomial por um algoritmo determinista (ou seja, o problema pertence a classe P) pode-se afirmar que, na prática, é viável calcular uma solução para ele com este algoritmo.

Questão 2 _(0,8) Considere um problema de otimização, A , que pode ser resolvido em $O(x)$ e um outro problema de decisão, B , que pode ser resolvido em $O(y)$. Compare as complexidades dos problemas A e B apresentados usando a notação assintótica adequada e explique por que é possível relacioná-los de tal forma (ou seja, qual a relação entre eles).

Questão 3 _(1,0) Um fato curioso em determinados problemas é que um pequeno detalhe pode fazer muita diferença em termos de viabilidade computacional. Considere o problema do CNF-SAT, por exemplo: a versão 2-CNF-SAT, que possui apenas duas variáveis por cláusula é um problema que pode ser resolvido de forma eficiente (em tempo polinomial) por um algoritmo determinista. Por outro lado, a versão 3-CNF-SAT já se enquadra como um problema NP-Completo e, portanto, não se conhece uma solução determinista eficiente. Agora, se considerarmos a versão C^n -CNF-SAT, onde cada cláusula possui C^n variáveis, esse problema ainda pertence à classe de problemas NP-Completo, ou passa a se enquadrar em outra classe de problemas (qual)? Elabore e justifique sua resposta usando a ideia de certificado.

Questão 4 ^(1.6) (a)_{1.0} Realize uma redução da seguinte instância do problema CLIQUE abaixo para uma instância do problema SUBSET-SUM. Para tanto, faça a conversão intermediária de CLIQUE \Rightarrow 3-CNF-SAT \Rightarrow SUBSET-SUM. Considere os vértices pintados como variáveis negadas e os vértices não pintados como variáveis não negadas.



(b)_{0.6} Para a instância do 3-CNF-SAT, elabore uma solução válida (resultado = verdade) e outra não válida (resultado = falsidade) e as soluções correspondentes na instância do SUBSET-SUM.

Questão 5 ^(1.3) Os departamentos da UDESC estão com uma grande dificuldade em montar o cronograma dos horários das aulas para o ano que vem, visto que é necessário considerar horários de disponibilidade dos professores, preferências de horários, prevenção de choques de horários entre disciplinas de um mesmo semestre ou disciplinas ministradas por um mesmo professor, minimização de choques de horários entre disciplinas de fases próximas, entre muitas outras restrições. Realmente, elaborar um cronograma ótimo é um problema muito difícil de ser resolvido. Considerando que este problema é NP-Completo, um aluno da turma de complexidade argumentou que a melhor solução provavelmente seria usar um algoritmo de *backtracking*, ao passo que o outro rebateu dizendo que uma heurística seria melhor. Sobre a discussão iniciada, responda:

- ^(0.8) Você concorda com alguma das propostas. Em caso positivo, justifique sua resposta argumentando por que a proposta escolhida é melhor do que a outra; em caso negativo, apresente a sua proposta e argumente por que sua nova proposta é melhor.
- ^(0.5) Considerando que um dos colegas sugeriu a utilização de uma heurística, diga qual a diferença entre uma abordagem heurística e um algoritmo de aproximação.

Questão 6 ^(1.5) Uma empresa de extração de pedras preciosas recebeu um visto para extrair 5 toneladas desse material numa região da Amazônia (será que teve propina envolvida no meio?!). Como é uma autorização para uma única extração, a empresa quer aproveitar ao máximo a oportunidade, para gerar a carga mais lucrativa possível. Considerando o seguinte relatório de disponibilidade das pedras, quais são os melhores minerais a serem extraídos e transportados? Considere que é possível quebrar os minerais em várias partes, mas ao fazer isso, o material cortado tem seu valor proporcional reduzido pela metade.

(ex: 3 tons de diamante possuem o valor de R\$ 600k. Ao extrair uma tonelada apenas, seu valor proporcional é R\$ 200k, mas por ter sido cortado, seu real valor será de R\$ 100k apenas)

	Ametista	Âmbar	Citrino	Diamante	Esmeralda	Ruby	Safira
Disponibilidade (tons)	1,6	2,8	1,9	0,7	1,1	1,2	2,5
Valor (R\$)	416k	406k	361k	350k	319k	312k	800k

a) ^(0.6) Para este problema, qual a melhor abordagem a ser usada para obter a resposta: Divisão e Conquista, Algoritmo Guloso, Algoritmo de *Backtracking* ou Programação Dinâmica? Por quê? A qual problema visto em aula podemos associá-lo?

b) ^(0.9) Apresente uma possível solução ótima ^(0.5) e responda se existe mais de uma solução ótima, justificando sua resposta?