questão 1

Dizemos que um algoritmo resolve um problema em tempo polinomial quando se sua eficiência de pior caso pertence a O(p(n)) onde p(n) é um polinomio de tamanho de entrada do problema n.

questão 2

Podemos dizer que um problema é tratável quando ele pode ser resolvido em tempo polinomial.

Questão 3

Podemos dizer que um problema não é tratável quando ele não pode ser resolvido em tempo polinomial.

Questão 4

A classe P é uma classe de problemas de decisão que podem ser resolvidos em

tempo polinomial por algoritmos (determinísticos). Essa classe de problemas é chamada

polinomial

Questão 5:

Podemos provar que um problema pertence a classe P se conseguirmos construir um algoritmo deterministico que resolve esse problema em um tempo polinomial.

questão 6:

Um problema decidível é quando é possível contruir um algoritmo que responda sim ou não se o problema pertence a um determinado conjunto. O problema indecidivel é quando não é possível contruir esse algoritmo.

questão 7

um algoritmo determinístico é um algoritmo em que, dada uma certa entrada, ela produzirá sempre a mesma saída, com a máquina responsável sempre passando pela mesma sequência de estados.

questão 8

é um algoritmo em que, dada uma certa entrada, pode apresentar comportamentos diferentes em diferentes execuções, ao contrário de um algoritmo determinístico.

questão 9

A classe NP é a classe de problemas de decisão que podem ser resolvidos por

algoritmos polinomiais não determinísticos. Essa classe de problemas é chamada de polinômio não-terminístico.

questão 11

Se um problema está em P, é possível utilizar o algoritmo de tempo polinomial que resolve no estágio de verificação do algoritmo não deterministico que simplismente ignorando a suposição.

questão 12

Se P = NP implicaria que muitos problemas de decisão combinatória poderia ser resolvido por um algoritmo de tempo polinomial.

questão 13

NP-completo é um subconjunto de NP, o conjunto de todos os problemas de decisão cujas soluções podem ser verificadas em tempo polinomial. Para provar que um problema é NP-completo, basta formula-lo como um problema NP-completo já conhecido.

questão 14.

Dado um problema, para provar que ele é NP-completo, basta formula-lo como outro problema NP-completo já conhecido.

questão 15

Significa que NP-completo está contido na classe NP, em ?????

questão 16

????

questão 17

e

classe P

classe de problemas de decisão que pode ser resolvido em tempo polinomial por algoritmos determinísticos.

Classe NP

classe de problemas de decisão que pode ser resolvido em tempo polinomial por algoritmos não determinísticos.

Não Determinísticos: Um procedimento que recebe uma entrada I e faz

1. Gera uma solução candidata S para cada instância I
2. Dado I e S, retorna sim se S é uma solução de I (tempo polinomial)

Classe NP-Completo

Um problema de decisão D é NP-completo se

1. Pertence a classe NP
2. Todo problema da classe NP é polinomialmente redutível a D

Um problema de decisão D1 é dito polinomialmente redutível à D2 se existe uma função t que transforma instâncias de D1 em instâncias de D2 de forma que:

D t mapeia todas as instâncias sim de D1 em instâncias Sim de D2 e todas as instâncias não de D1 em instâncias não de D2

t é computável em tempo polinomial.