Prova **

CODCRED - DISCIPLINA - TURMA - SEMESTRE - Prof. Diego Noble

Nome: ____________________ Nota: ______

Instruções

- A prova é individual e sem consulta a qualquer material.
- Somente respostas escritas à caneta serão reavaliadas.
- É proibido fazer perguntas de interpretação da prova após a leitura da mesma pelo professor.
- É proibido o empréstimo de material durante a prova.
- Só é permitida a entrega da prova transcorridos 20 minutos do seu início.
- A prova tem valor total de 12.5 pontos.

Questão 1 *Algoritmos Gulosos*

_____/ 2 p.

Você precisa definir a posição de bases de celular para cobrir *todas* as casas que ficam ao longo de uma estrada. A estrada pode ser interpretada como um segmento de reta com pontas leste e oeste. Cada base de celular cobre 4 quilômetros para cada lado. Dê um algoritmo eficiente para encontrar todas as casas ao longo da estrada usando o menor número de bases possível.

Questão 2 *Divisão-e-conquista*

_____/ 3 p.

Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ e $T(n)$ funções definidas sobre inteiros não-negativos, temos que:

$$T(n) = aT(n/b) + f(n)$$

Então $T(n)$ pode ser limitado assintoticamente por três regras:

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para uma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$.
2. Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para uma constante $\epsilon > 0$ e se $af(n/b) \leq cf(n)$ para uma constante $c < 1$, então $T(n) = \Theta(f(n))$.

Use o Método Mestre para encotrar o limite assintótico das seguintes relações de recorrência:

(a) $T(n) = 2T(n/2) + n^4$
(b) $T(n) = T(7n/10) + n$
(c) $T(n) = 16T(n/4) + n^2$
(d) $T(n) = 7T(n/3) + n^2$
(e) $T(n) = 7T(n/2) + n^2$
(f) $T(n) = 8T(n/2) + n^3$

Questão 3 *Algoritmos Gulosos*

_____/ 2.5 p.

Considere o problema de encontrar o troco para n centavos usando o menor número possível de moedas. Assuma que cada moeda tem um respectivo valor inteiro.

(a) Descreva um algoritmo guloso em pseudo-código que determina o troco considerando quatro valores de moedas: 25, 10, 5 e 1 centavos. Justifique com suas palavras porque o algoritmo encontra o troco correto usando o menor número de moedas possível para este conjunto de valores.

(b) Agora suponha que os valores das moedas disponíveis são um produto de um fator inteiro $c > 1$. Por exemplo, supondo $c = 2$, temos os seguintes valores: 2^0, 2^1, 2^2, ..., 2^k, onde $k \geq 1$. Mostre que o seu algoritmo também encontra o troco usando um número mínimo de moedas neste caso.

(c) Dê um conjunto de valores para as moedas de forma com que o seu algoritmo guloso não encontre a resposta ótima. Você deve incluir necessariamente a moeda de 1 centavo neste conjunto. Justifique a resposta.

Questão 4

Considere o seguinte código abaixo em Java:

___/ 3 p.

```java
import java.util.Random;
class Point{
    private double x,y;
    public Point(double x, double y) { this.x = x; this.y = y;}
    public static double distance(Point a, Point b){
        double dx = a.x - b.x;
        double dy = a.y - b.y;
        return Math.sqrt(dx*dx+dy*dy);
    }
}
public class TSP{
    static boolean tspCertf(Point[] pts, double k, Integer[] sol) { }
    public static void main(String[] args){
        Point[] points = new Point[100];  // Vetor com a cidades
        Integer[] sol = new Integer[100]; // Vetor com a solução
        Random rg = new Random();
        Double x,y;
        for(int i = 0; i < points.length; i++){
            x = 100.0 * rg.nextDouble();
            y = 100.0 * rg.nextDouble();
            points[i] = new Point(x, y); // Cria/insere um Point no array
            sol[i] = i;  // sol é inicializado com valores de 0 até 99
        }
        sol.shuffleArray(); // Embaralha aleatoriamente os elementos de sol
        boolean ans = tspCertf(points, 1000.0, sol);
        System.out.println(ans);
    }
}
```

A classe `Point` modela pontos (x, y) em um sistema cartersiano e tem um método de classe para calcular a distância entre dois pontos. A classe `TSP` cria alguns pontos aleatórios como demonstração e tem um método `tspCertf`.

Questão 5

O pseudo-código abaixo para serve para realizar a busca em largura em grafo.

______/ 2 (+0.5) p.

Entrada: Um grafo $G = (V, E)$ e um vértice origem $s \in V$.

```
início
    para cada v ∈ V, v ≠ s faça
        v.dist ← ∞;
        v.visited ← false;
    s.dist ← 0;
    s.visited ← true;
    Q ← ∅;
    Q.ENQUEUE(s);
    enquanto Q ≠ ∅ faça
        u ← Q.DEQUEUE();
        para cada v ∈ G.Adj[u] faça
            se v.visited == false então
                v.visited ← true;
                v.dist ← u.dist + 1;
                Q.ENQUEUE(v);
        u.visited ← true;
```

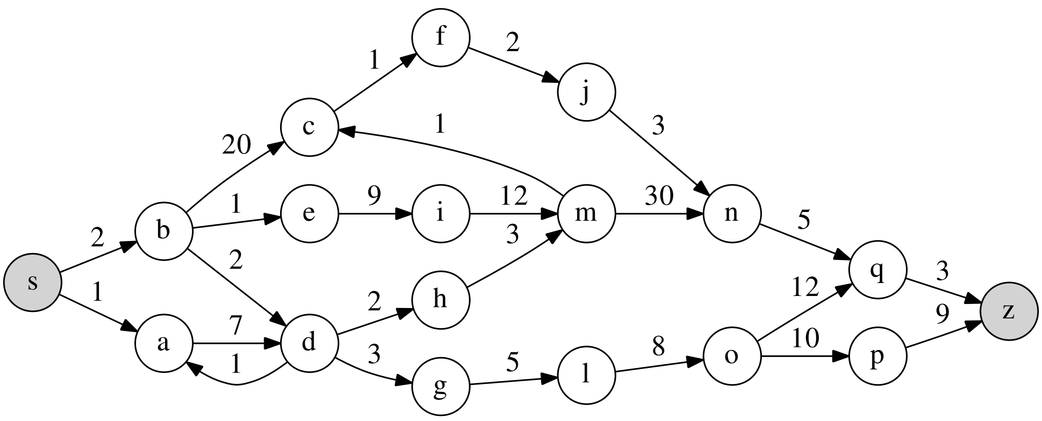

Figura 1: Um grafo.

Questão 6

Algumas afirmações abaixo são verdadeiras e outras são falsas:

(a) Morbi ac imperdiet nisi. Interdum et malesuada fames ac ante ipsum primis in faucibus. Donec lobortis magna nibh, a consequat libero convallis sit amet.
(b) Phasellus tellus libero, gravida tempor fringilla et, viverra ut sem. Nunc et nulla neque.
(c) Lisque id eros pellentesque, fringilla neque at, ultricies lorem.
(d) Integer a efficitur leo.
(e) Sed facilisis tempus tellus sit amet elementum.

Preencha a tabela abaixo com **V** para as respectivas afirmações verdadeiras e **F** nas respectivas afirmações falsas.

Afirmação	a	b	c	d	e
Resposta					

Resposta 6

Afirmação	a	b	c	d	e
Resposta	V	V	F	V	V