

Aluno:

Matricula:

**Obs:**

- Nesta prova assuma que o uso de “includes” e de diretivas do tipo “using namespace std;” não seja necessário.
- A não ser que a questão afirme de forma explícita o contrário, assuma que os grafos são simples
- Diferentemente de C++, a ordem com que funções são declaradas não importa (ou seja, se uma função  $x()$  chama  $y()$   $\rightarrow y()$  não precisa ser declarada antes de  $x()$  )

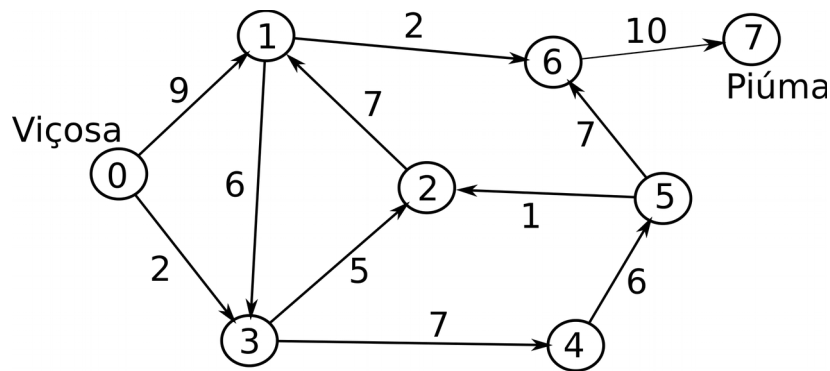
-----  
Nesta prova você poderá chamar de maneira “informal” qualquer um dos algoritmos vistos em sala (mencione o nome do autor do algoritmo e detalhes como, por exemplo, onde está a entrada do algoritmo e onde/como a saída será gravada)

Por exemplo, o código abaixo seria aceito nesta prova:

```
vector<vector<pair<int,int> > > adj(n); //lista de adjacencia com pesos...
vector<int> distancia(n);
.... // suponha que aqui tenha o codigo para preencher a lista de adjacencia
int v;
cin >> v;
//Use o algoritmo de Dijkstra na lista de adjacencia “adj” (onde cada par contém o id do vértice vizinho e
o peso da aresta correspondente) para preencher “distancia” com a distancia entre o vértice v e todos os
outros
cout << “Distancia entre “ << v << “ e o vértice “ << 5 << “ : “ << distancia[5] << endl;
```

-----

1) (valor: 7) Os alunos da UFV (a maioria mineiros) decidiram organizar uma viagem para a praia de Piúma durante as férias de verão. Para facilitar o planejamento da viagem, criaram um digrafo onde cada cidade é representada por um vértice e as estradas (direcionadas) são representadas por arestas (sendo que as estradas se conectam apenas nas cidades). Porém, um terrorista teve acesso ao grafo e decidiu atrapalhar a viagem explodindo bombas em várias estradas de modo a deixar impossível viajar de Viçosa (vértice 0) para Piúma (vértice 7). Para planejar quais estradas explodir, o terrorista colocou um peso em cada aresta, sendo que tal peso representa o custo das bombas necessárias para impossibilitar o tráfego naquela estrada.



a) (valor: 1) Mostre qual o custo total mínimo das estradas que devem ser destruídas para impossibilitar o tráfego de Vicos a Piúma. Marque no grafo acima quais arestas deverão ser destruídas. (não é necessário justificar)

b) (valor: 4) Como poderia ser implementado um algoritmo para encontrar o custo para impedir o fluxo de uma cidade A para a cidade B? Se sua solução envolver o uso de algum algoritmo já pronto, cite o nome do algoritmo (ou seja, não diga apenas algo como “encontre uma AGM no grafo criado fazendo ...”, mas sim “encontre uma AGM em G usando o algoritmo de Prim no grafo criado fazendo ...”)

c) (valor: 2) Suponha que o terrorista deseja não apenas desconectar o tráfego de uma cidade A para outra cidade B, mas sim de A para um conjunto de cidades (por exemplo, ele pode querer deixar impossível viajar de Viçosa para o Rio de Janeiro, para Belo Horizonte e para Teixeira). Como poderíamos encontrar o custo mínimo de se fazer esse atentado?

2) (valor: 7) Considere o problema Freekles da lista de exercícios (mas suponha que em cada execução você deverá resolver apenas uma instância do problema): nele, a entrada consiste em um conjunto de

pontos (em 2D) e o objetivo é ligar todos os pontos usando uma caneta de modo a minimizar a quantidade de tinta utilizada (ou seja, você deverá conectar os pontos usando linhas de modo a minimizar o comprimento total das linhas utilizadas).

Mais especificamente, a entrada começará com um número  $N$ . A seguir, haverá  $N$  linhas onde cada linha contém a coordenada  $(x,y)$  de um ponto em 2D. A saída deverá ser o comprimento total mínimo das linhas necessárias para conectar todos os pontos.

Exemplo de entrada:

```
3
1.0 1.0
2.0 2.0
2.0 4.0
```

Saída:

```
3.41
```

Explicação: ligue o ponto 1 no 2 (utilizando uma linha com aproximadamente 1.41 unidades de comprimento). Então, ligue o ponto 2 no 3 (utilizando uma linha com 2 unidades de comprimento). Não há forma de ligar os pontos usando menos do que 3.41 unidades de comprimento no total.

Implemente um programa em C++ que lê a entrada no formato descrito acima e imprime a saída conforme especificado (o formato da saída (em termos de número de casas de precisão) não é importante). Assuma que o valor de  $N$  usado nos testes seja no máximo 1000.

Observação: seu código pode “chamar” qualquer algoritmo visto em sala sem implementá-lo. Por exemplo, se você precisar resolver o problema do caixeiro viajante dentro do seu programa e tivermos visto em sala que o algoritmo de Arthur-Bernardes resolve isso, escreva no seu programa (como um comentário no meio do código C++) algo como “\\* Use o algoritmo de Arthur-Bernardes para resolver o problema do Caixeiro Viajante no grafo armazenado na matriz de adjacência adj e armazene no vetor result a sequência de vértices no ciclo encontrado pelo algoritmo \*\” (você só pode utilizar essas descrições em alto nível nas chamadas a algoritmos -- o resto do programa deverá ser implementado normalmente em C++)

**(1 ponto) Indique aqui a ordem de complexidade do seu algoritmo (em termos de  $N$ ):**

**Implementação:**





4) (valor: 6) Um carro híbrido possui um motor de combustão interna e um elétrico e utiliza várias técnicas para economizar combustível. Uma delas é chamada de frenagem regenerativa e consiste em recarregar a bateria com a energia da frenagem (basicamente o motor elétrico é utilizado como um gerador de energia -- a resistência criada ao se gerar a eletricidade propicia a frenagem).

Assim, o veículo pode gastar energia (vinda do combustível ou da bateria) para subir um morro e ganhar energia ao descer (obviamente, devido às perdas, a energia recuperada ao descer é sempre menor do que a gasta para subir um morro).

Uma empresa de automóveis decidiu criar um digrafo representando um mapa do Brasil, onde os vértices são cidades e as arestas (direcionadas) são estradas conectando cidades. Foi adicionado um peso às arestas, que representa o gasto (ou ganho) energético médio de se percorrer completamente a estrada correspondente (um peso 30 indica que se gasta 30 KWh de energia para percorrer a estrada e um peso -20 indica que a estrada, provavelmente com muita descida, fornecerá 20 KWh de energia para o carro que percorrê-la).

Observação:

-- A quantidade de cidades no mapa é relativamente grande (exemplo: 4,000 cidades) e a quantidade de estradas é relativamente similar à quantidade de cidades (exemplo: 12,000 estradas)

Você foi contratado para criar um software (que será adicionado a um GPS) que deverá calcular a rota mais vantajosa (energeticamente) para se viajar entre duas cidades do mapa.

a) (valor: 4) Qual dos algoritmos abaixo seria mais adequado para resolver o problema acima?

- a) Strassen
- b) Dijkstra
- c) Kruskal
- d) Floyd-Warshall
- e) Edmond-Karp
- f) Bellman-Ford-Moore
- g) Briot-Ruffini
- h) Prim

Justifique sua resposta:

b) (valor: 2) Um usuário utilizou seu software para planejar uma rota de Viçosa para Belo Horizonte. Após viajar por várias horas (usando uma rota que passou pelo Rio de Janeiro) ele percebeu que simplesmente minimizar o gasto energético pode não ser uma boa ideia (já que as rotas mais econômicas podem ser muito longas -- apesar disso raramente ocorrer visto que o atrito gasta muita energia e, assim, quando mais longa uma viagem mais energia ela normalmente gasta). O que você faria para que a qualidade das rotas obtidas pelo seu algoritmo seja melhor? (de modo que elas economizem energia e ao mesmo tempo sejam relativamente curtas -- não necessariamente usando o caminho mínimo)

