

LAB 04

MC558A

Data de Entrega: 08/10

A prefeitura da cidade de CincoCincoOitolândia está em apuros. Um enorme protesto de pais e responsáveis criticou duramente a distribuição dos alunos das escolas publicas do município. Alguns pais ficaram furiosos por terem que matricular seus filhos em escolas muito distantes de suas casas.

Para resolver o impasse, foi negociado que a prefeitura divulgaria uma nova distribuição dos alunos. Agora, cada aluno seria designado para a escola mais próxima de sua casa, ou pelo menos, a escola que tivesse um ponto de ônibus mais próximo a sua residência.

O grande problema é que CincoCincoOitolândia tem centenas de escolas e milhares de alunos. Como calcular a nova distribuição em tão pouco tempo?

Sua função será a de projetar um algoritmo para atribuir cada aluno a uma escola, respeitando a regra imposta. Lembre-se que a cidade é enorme e tem muitas escolas e alunos, então use tudo que sabe para deixar o seu código o mais rápido possível.



Existem diversas escolas espalhadas pela cidade, algumas dessas escolas possuem ônibus que transportam gratuitamente seus alunos, basta ir a um dos pontos de ônibus que a escola indica.

Para os responsáveis não há diferença entre deixar a criança na escola ou no ponto de ônibus da escola, eles preferem o que for mais próximo de casa. Caso existam duas ou mais escolas/pontos de ônibus com a mesma distância, os responsáveis sempre preferem matricular a criança na escola que vier primeiro na lista.

Você receberá o mapa da cidade (pontos de interesse e ruas), a indicação da localização das escolas, dos pontos de ônibus de cada escola e do endereço de cada um dos alunos. Seu programa deve informar quantos alunos vão estudar em cada escola e apresentar a relação de alunos matriculados em cada escola.

Você não precisa se preocupar com limites ou capacidades das escolas ou dos ônibus, isso será resolvido em outro momento.

Descrição da entrada :

V A
 $\{v_i \ v_j \ d_{ij}\}$
 $\{v_i \ v_j \ d_{ij}\}$
 \dots
 E
 $\{v_i \ o_i \ p_k^1 \ p_z^2 \ \dots \ p_w^{o_i}\}$
 $\{v_i \ o_i \ p_k^1 \ p_z^2 \ \dots \ p_w^{o_i}\}$
 \dots
 S
 $s_i \ v_k$
 $s_i \ v_k$

Dois inteiros V e A , representando o número de vértices (locais) e arestas (ruas) do mapa da cidade.

Seguido de A linhas no formato :

$\{v_i \ v_j \ d_{ij}\}$

Representando a distância entre os pontos i e j (a distância de v_i a v_j é igual a de v_j a v_i , $d_{ij} = d_{ji}$).

Um inteiro E representando um número de escolas, seguido de E linhas no formato:

$\{v_i \ o_i \ p_k^1 \ p_z^2 \ \dots \ p_w^{o_i}\}$

Respectivamente o vértice onde está a escola, quantos pontos de ônibus ela possui e qual o vértice de cada um dos pontos de ônibus.

Por fim, um inteiro S representando o número de estudantes, seguido de S linhas com o localização (o vértice onde está localizado) a casa do i -ésimo estudante.

Um mesmo vértice pode eventualmente indicar uma ou mais escolas, pontos de ônibus e localização de alunos (Sim, quebramos as regras da física aqui.), a distância de um vértice a ele mesmo é 0.

Limites :

$$1 \leq V, E, S \leq 10^5$$

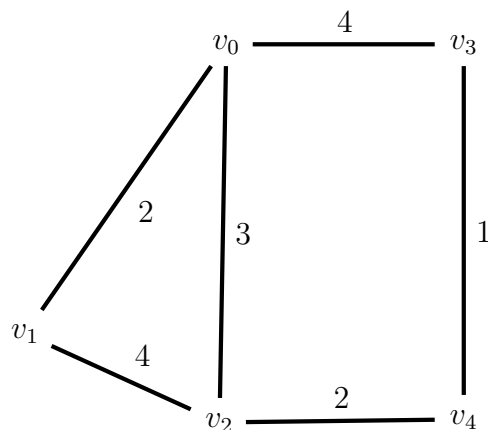
$$1 \leq A \leq 10^6$$

$$1 \leq d_{ij} \leq 10$$

Tempo Limite : 1s

Descrição da saída :

Para cada escola deve-se apresentar um inteiro informando o número de alunos cadastrados, seguido de forma ordenada o vértice onde está localizado cada um desses alunos.

Exemplo 1**Entrada**

```

5 6
0 1 2
0 2 3
0 3 4
1 2 4
2 4 2
3 4 1
2
0 0
4 0
3
1
2
3

```

Saída

```

1
1
2
2
3

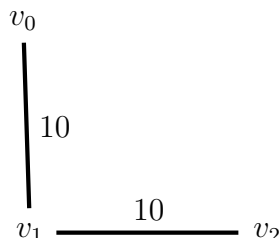
```

Tabela 1: Exemplo 1

Análise Exemplo 1 Temos os vértices v_0 e v_4 com escolas, ambos com 0 pontos de ônibus, e alunos nos pontos 1, 2 e 3. Note que a distancia mínima do aluno no vértice v_1 para a primeira escola é de 2, e a distancia para a escola no vértice v_4 é de 6, logo o aluno de v_1 estudará na primeira escola. Os alunos em v_2 e v_3 estão mais proximos da escola localizada em v_4 , assim, vão estudar nessa escola.

A saída informa que a primeira escola tem 1 aluno, e informa que este é o aluno do vértice 1. A segunda escola tem 2 alunos, os dos vértices 2 e 3.

Exemplo 2



Entrada

```

3 2
0 1 10
1 2 10
2
0 0
2 0
1
1

```

Saída

```

1
1
0

```

Tabela 2: Exemplo 2

Análise Exemplo 2 Esse exemplo visa demonstrar os critérios de desempate, o aluno localizado em v_1 está a distancia 10 da escola de v_0 e da escola de v_2 , mas como a escola de v_0 foi apresentada primeiro na lista de entrada, ela será preferida pelo aluno. Assim, na saída, a primeira escola terá 1 aluno, que é o aluno 1, e a segunda escola fica sem alunos.

Observações e Avaliação:

- Os programas que não estiverem compilando ou não passarem em algum dos testes pré-instalados no SuSy, terão nota 0.
- O arquivo fonte deve estar bem comentado! Qualquer função ou trecho de código não trivial deve conter uma breve descrição sobre o seu propósito.
- No inicio do arquivo fonte enviado ao SUSY, deverá haver uma descrição em alto nível sobre como a solução foi pensada e implementada. Também é necessário justificar a complexidade do algoritmo. É esperado uma descrição com cerca de 100 a 200 palavras.
- Um esqueleto da aplicação está disponível na pagina da disciplina, o uso é obrigatório. Entrada e saída de dados já estão disponíveis na main. É necessário somente implementar a função *solve()*, você pode implementar também outras funções auxiliares caso julgue necessário. Para compilar, basta executar o comando *make* no endereço do fonte, para testar com o exemplo, basta executar *make run*.

- É esperado que esta atividade demande cerca de 4 a 8 horas para ser finalizada, caso esteja demorando muito mais tempo, ou caso tenha qualquer dúvida, procure atendimento o quanto antes!
- Em caso de plágio, todos os alunos envolvidos serão imediatamente reprovados.