PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Disciplina	Curso	Turno	Período
Algoritmos em Grafos	Ciência da Computação	Tarde	4°

Professor
Felipe Cunha (felipe@pucminas.br)

Eu acredito, que as vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.
Alan Turing

Trabalho Prático 01

Data de Entrega: **02/05/2018**
Valor: **8pts**

Resolva os problemas listados abaixo individualmente, usando a linguagem c++. Ao final, submeta na plataforma verde e apresente a solução ao professor na data agendada.

Colônia de Formigas

Um grupo de formigas está muito orgulhoso pois construíram uma grande e magnífica colônia. No entanto, seu enorme tamanho tem se tornado um problema, pois muitas formigas não sabem o caminho entre algumas partes da colônia. Elas precisam de sua ajuda desesperadamente!

A colônia de formigas foi criada como uma série de **N** formigueiros conectados por túneis. As formigas, obssessivas como são, numeraram os formigueiros sequencialmente à medida que os construiam. O primeiro formigueiro, numerado 0, não necessitava nenhum túnel, mas para cada um dos formigueiros subsequentes, 1 até **N-1**, as formigas também construíram um único túnel que conectava o novo formigueiro a um dos formigueiros existentes. Certamente, esse túnel era suficiente para permitir que qualquer formiga visitasse qualquer formigueiro já construído, possivelmente passando através de outros formigueiros pelo percurso, portanto elas não se preocupavam em fazer novos túneis e continuavam construindo mais formigueiros.

O seu trabalho é: dada a estrutura de uma colônia e um conjunto de consultas, calcular, para cada uma das consultas, o menor caminho entre pares de formigueiros. O comprimento do caminho é a soma dos comprimentos de todos os túneis que necessitam ser visitados.

Entrada

Cada caso de teste se estende por várias linhas. A primeira linha contém um inteiro **N** representando a quantidade de formigueiros na colônia ($2 \leq$ **N** $\leq 10^5$). Cada uma das próximas **N-1** linhas contém dois inteiros que descrevem um túnel. A linha *i*, para $1 \leq i \leq$ **N-1**, contém **A*i*** e **L*i***, indicando que o formigueiro *i* foi conectado diretamente ao formigueiro **A*i*** por um túnel de comprimento **L*i*** ($0 \leq$ **A*i*** $\leq$ **i-1** e $1 \leq$ **L*i*** $\leq 10^9$). A próxima linha contém um inteiro **Q**representando o número de consultas que seguem ($1 \leq$ **Q** $\leq 10^5$). Cada uma das **Q** linhas seguintes descreve uma consulta e contém dois inteiros distintos **S** e **T** ($0 \leq$ **S**,**T** $\leq$ **N-1**), representando, respectivamente, os formigueiros de origem e destino.

O último caso de teste é seguido por uma linha contendo apenas um zero.

Saída

Para cada caso de teste, imprima uma única linha com **Q** inteiros, os comprimentos do menor caminho entre os dois formigueiros de cada consulta. Escreva os resultados para cada consulta na mesma ordem em que aparecem na entrada.

Exemplo de Entrada	Exemplo de Saída
6 0 8 1 7 1 9 0 3 4 2	16 20 11 17 1 1 5000000000

Exemplo de Entrada	Exemplo de Saída
4 2 3 5 2 1 4 0 3 2 0 1 2 1 0 0 1 6 0 1000000000 1 1000000000 2 1000000000 3 1000000000 4 1000000000 1 5 0 0	

Fibra Óptica

Um país em desenvolvimento está tentando melhorar sua infra-estrutura de comunicação. Atualmente, cada cidade do país tem a sua rede de computadores local, mas não há uma comunicação rápida entre as cidades. O Ministério Autônomo das Comunicações (ACM) do país decidiu criar uma rede de fibra óptica de grande velocidade que ligará todas as cidades. A fim de fazer isso, eles decidiram fazer a seguinte abordagem. Pares de cidades foram escolhidas para ter uma ligação de fibra óptica instalada entre elas. A escolha foi tal que haverá apenas um caminho de fibra entre qualquer par de cidades, a fim de reduzir o custo. Os pares de cidades foram escolhidos considerando diversos fatores, incluindo a análise de demanda estimada e a distância entre as cidades.

Cada cidade terá um roteador óptico instalado, o qual será utilizado para conectar todas as ligações óticas com uma extremidade da cidade. Em cada cidade, há muitos locais diferentes onde o roteador óptico pode ser instalados. Sua tarefa, como engenheiro que está trabalhando neste projeto, é desenvolver um programa de computador que receba as localizações de cada uma das cidades e minimize o tamanho total de fibra que seria necessária para este projeto.

Entrada

A entrada consiste de vários casos de teste. Cada caso de teste começa com uma linha contendo o número de cidades **N** (1 ≤ **N** ≤ 1000) no país. A seguir, para cada cidade, há uma sequência de linhas. A primeira linha contém o nome (único) da cidade (apenas letras maiúsculas, no máximo de 15 letras), e o número de locais candidatos **Ci** (1 ≤ **Ci** ≤ 50) em que o roteador óptico pode ser instalado. Então, existe uma linha para cada local candidato, contendo dois inteiros **X** e **Y** que representam as coordenadas do local (-10000 ≤ **X**, **Y** ≤ 10000). Você deve usar a distância euclidiana entre os sites para calcular o comprimento da fibra correspondente necessário para ligá-los. Depois da descrição de cada cidade, com seus sites candidatos, haverá **N** - 1 linhas, cada uma delas contendo o nomes de duas cidades que terão um link de fibra instalado entre elas. O final da entrada é indicado por **N** = 0.

Saída

Para cada caso de teste, seu programa deve imprimir uma linha com o comprimento total mínimo de fibra óptica necessária para ligar as cidades informadas. Sua resposta deve ser arredondada para um dígito decimal.

Exemplo de Entrada	Exemplo de Saída
3 AUSTIN 1 500 500 DALLAS 2 1000 10 990 -10 ELPASO 2 0 0 30 0	1646.3 189.9

Exemplo de Entrada	Exemplo de Saída
ELPASO AUSTIN DALLAS ELPASO 3 HUSTON 3 100 0 100 50 100 100 AUSTIN 2 200 0 180 40 SANANTONIO 2 0 -10 10 -50 HUSTON AUSTIN HUSTON SANANTONIO 0	

Back to the Future

Um grupo de amigos resolveu ir à Alemanha para apoiar a seleção brasileira em sua jornada gloriosa rumo ao hexa. Como as passagens aéreas e as estadias eram caras, cada um trouxe uma quantidade de dinheiro que julgou suficiente para passar o mês com conforto e voltar para casa sem problemas.

Porém, após a bela campanha do Brasil na copa do mundo, o grupo de amigos se viu obrigado a gastar o dinheiro que tinha guardado para as etapas finais da copa com a famosa cerveja alemã. As consequências de tais atos foram terríveis. Após uma grande bebedeira, todos foram pegos pela polícia local dormindo na rua, e receberam multas pesadíssimas. Além disso, todos perderam suas passagens de volta. Devido a esses contratempos, a viagem de volta ficou ameaçada. De repente, eles descobriram que precisavam voltar para casa gastando a menor quantidade possível de dinheiro.

Analisando as rotas aéreas disponíveis, os amigos notaram que em todas as rotas o número de assentos disponíveis nos aviões era sempre o mesmo. Porém, os preços das viagens entre uma cidade e outra eventualmente variavam bastante. Assustados com a possibilidade de não encontrar lugares suficiente nos aviões para que todos pudessem voltar e preocupados em gastar a menor quantidade possível de dinheiro, o grupo de amigos resolveu pedir sua ajuda.

Entrada

O problema é composto por várias instâncias. Cada instância começa com uma linha com dois inteiros positivos **N**($2 \leq$ **N** ≤ 100) e **M** ($1 \leq$ **M** ≤ 5000), onde **N** é o número de cidades que pertencem às **M** rotas de voo consideradas. Os amigos querem ir da cidade 1 até a cidade **N**.

Nas próximas **M** linhas são fornecidos triplas de inteiros **A B C** descrevendo a rota do avião (**A** e **B**) e o preço da passagem aérea por pessoa (**C**). Os valores de **A** e **B** estão entre 1 e n. As rotas são bidirecionais (ou seja, há um voo de **A** até **B** e um voo de **B** até **A** com preço **C**) e haverá no máximo uma rota entre duas cidades. Na próxima linha são dados dois inteiros, D e K, onde D é o número de amigos e K é o número de assentos livres em cada voo. Cada rota só pode ser utilizada uma vez.

Saída

Para cada instância, imprima a linha "Instancia **k**", onde **k** é o número da instância atual. Além disso, imprima a menor quantidade possível de dinheiro que os amigos vão gastar para voltar ao Brasil (que está limitada por 10^{15}). Caso não seja possível escolher um conjunto de voos que levem todos para casa, imprima "impossivel".

Imprima uma linha em branco após cada instância.

Exemplo de Entrada	Exemplo de Saída
4 5 1 4 1 1 3 3 3 4 4	Instancia 1 80 Instancia 2

Exemplo de Entrada	Exemplo de Saída
1 2 2 2 4 5 20 10 4 4 1 3 3 3 4 4 1 2 2 2 4 5 20 100 4 4 1 3 3 3 4 4 1 2 2 2 4 5 20 1	140 Instancia 3 impossivel

Colorindo Grafos

Seja G um grafo simples com N vértices coloridos e M arestas. Nós desejamos saber se é possível adicionar exatamente P arestas em G de tal forma que o grafo resultante seja simples, conexo e nenhuma aresta conecte dois vértices da mesma cor.

Entrada

A entrada contém múltiplos casos testes. A primeira linha contém a quantidade de casos testes **T** (**T** < 70). Cada caso teste começa com 4 inteiros na seguinte ordem: o número de vértices **N** (1 <= **N** <= 10^3), o número de arestas no grafo original **M** (0 <= **M** <= 10^5), o número de arestas a serem inseridas **P** (0 <= **P** <= 10^6) e o número de cores **K** (1 <= **K** <= 10^3). A linha seguinte contém **N** números **Xi** indicando a cor do i-ésimo vértice (1 <= **Xi** <= **K**). As **M** seguintes linhas contém um par de inteiros (**V_i**, **V_j**) indicando a presença de uma aresta entre os vértices **V_i** e **V_j**. (1 <= **V_i**,**V_j** <= **N**).

Saída

Para cada caso teste, imprima uma única lina com "**Y**" (sem aspas) se é possível construir tal grafo ou "**N**" caso contrário.

Exemplo de Entrada	Exemplo de Saída
2 4 2 1 2 1 1 2 2 1 3 2 4 4 1 1 2 1 1 2 2 1 3	Y N