Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Ciência da Computação
Algoritmos em Grafos
Prof^{a}. Raquel Mini

Trabalho Prático

Data de Entrega: 27/10/2019
Valor: 20 pontos

Resolva os problemas listados abaixo **individualmente**, usando a linguagem C++. Ao final, submeta na plataforma verde.

Componentes Conexos

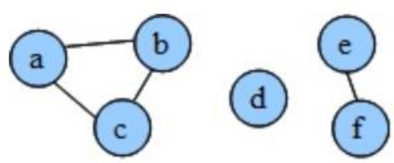
Com base nestas três definições:

Grafo conexo: Um grafo **G(V,A)** é conexo se para cada par de vértices *u* e *v* existe um caminho entre *u* e *v*. Um grafo com apenas um componente é um grafo conexo.

Grafo desconexo: Um grafo **G(V,A)** é desconexo se ele for formado por 2 ou mais componentes conexos.

Componente conexo: Componentes conexos de um grafo são os subgrafos conexos deste grafo.

O grafo a seguir possui 3 componentes conexos. O primeiro é formado pelos vértices *a, b, c*. O segundo é formado unicamente pelo vértice *d* e o terceiro componente é formado pelos vértices *e, f*.



Com base nestes conceitos, onde cada entrada fornecida que tem a identificação de cada um dos vértices, arestas e as ligações entre os vértices através destas arestas, liste cada um dos componentes conexos que existem no grafo, segundo a entrada fornecida.

Entrada

A primeira linha do arquivo de entrada contém um valor inteiro **N** que representa a quantidade de casos de teste que vem a seguir. Cada caso de teste contém dois valores **V** e **E** que são, respectivamente, a quantidade de **V**értices e arestas (**E**dges) do grafo. Seguem **E** linhas na sequência, cada uma delas representando uma das arestas que ligam tais vértices. Cada vértice é representado por uma letra minúscula do alfabeto (*'a'-'z'*), ou seja, cada grafo pode ter no máximo 26 vértices. Cada grafo tem no mínimo 1 componente conexo.

Obs: Os vértices de cada caso de teste sempre iniciam no 'a'. Isso significa que um caso de teste que tem 3 vértices, tem obrigatoriamente os vértices 'a', 'b' e 'c'.

Saída

Para cada caso de teste da entrada, deve ser apresentada uma mensagem **Case #n:**, onde **n** indica o número do caso de teste (conforme exemplo abaixo). Segue a listagem dos vértices de cada segmento, um segmento por linha, separados por vírgula (inclusive com uma virgula no final da linha). Finalizando o caso de teste, deve ser apresentada uma mensagem indicando a quantidade de componentes conexos do grafo (em inglês). Todo caso de teste deve ter uma linha em branco no final, inclusive o último caso de teste.

Obs: os nodos devem sempre ser apresentados em ordem crescente e se há caminho de a até b significa que há caminho de b até a.

Exemplo de Entrada	Exemplo de Saída
3 3 1 a c 10 10 a b a c a g b c c g e d d f h i i j j h 6 4 a b b c c a e f	Case #1: a,c, b, 2 connected components Case #2: a,b,c,g, d,e,f, h,i,j, 3 connected components Case #3: a,b,c, d, e,f, 3 connected components

Resgate em Queda Livre

Ó, meu Deus! Um grupo de pessoas está caindo em queda livre! Elas saltaram todas exatamente ao mesmo tempo de vários aviões que estavam exatamente à mesma altura. A intenção era realizar o maior e mais belo salto sincronizado da História. No entanto, o malévolo Loki, para se deleitar com a insignificância humana, sabotara os paraquedas, e agora a única esperança está numa ação conjunta do Homem-Aranha com o Homem-de-Ferro. Como ambos são muito nerds, notaram que as pessoas estavam caindo todas num mesmo plano paralelo ao solo, a despeito da resistência do ar e de outros fatores. Então, bolaram um plano infalível. Primeiro, o aracnídeo unirá todas as pessoas através de cabos de teia entre elas. Uma vez que não haja pessoa que não esteja conectada ao grupo, o playboy poderá eletromagnetizar o grupo todo e, segurando na mão de uma apenas das pessoas do grupo, pousar todas elas em segurança.

Mas não há muito tempo para divagações. O Homem-Aranha precisa agir rápido, o que no caso dele significa gastar o mínimo possível de teia. Para tanto, o Homem-de-Ferro em seu screen projetou numa malha cartesiana o plano em que as pessoas estão, usando o centímetro como unidade de medida, e obteve as coordenadas de cada pessoa na malha. Agora, J.A.R.V.I.S. está computando qual o mínimo necessário de teia de que o Homem-Aranha precisará. Dependendo da resposta, o Homem-de-Ferro não esperará pelo garoto e improvisará alguma outra peripécia.

Entrada

A entrada é constituída por vários casos de teste. A primeira linha de entrada contém um inteiro **C** que determina a quantidade de casos de teste. Cada caso de teste começa com um inteiro positivo **n** (**n** ≤ 500), o qual representa o número de pessoas no grupo. Seguem, então, **n** linhas, cada uma designando uma pessoa do grupo pelas suas coordenadas **x** e **y** na malha ($0 \leq \mathbf{x}, \mathbf{y} \leq 10^4$).

Saída

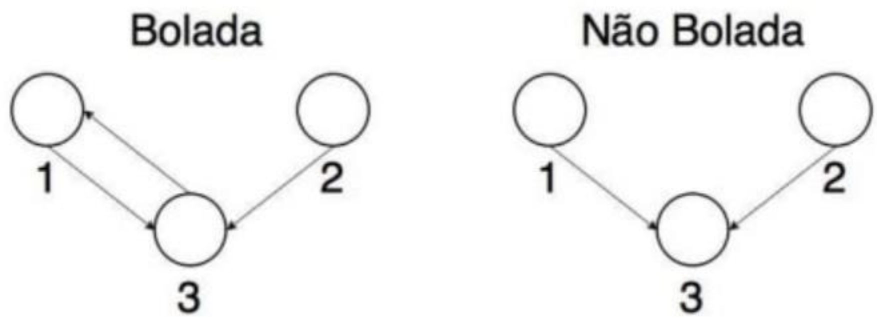
Para cada caso de teste, seu programa deverá imprimir uma linha contendo o valor com precisão de duas casas decimais correspondente ao comprimento mínimo de teia, em metros, necessário para se conectarem todas as pessoas do grupo. Atente para que o separador das casas decimais seja **.** (ponto), não **,** (vírgula).

Exemplo de Entrada	Exemplo de Saída
2 5 0 0 0 100 100 200 200 400 300 300 4 1 5 1 4 2 3 3 2	6.06 0.04

Cadeia Alimentar

Na viagem de Bino, ele desenhou algumas cadeias alimentares dos seres de Binox. Porém ele notou que em algumas cadeias, todos os pares de espécies se relacionavam (diretamente ou indiretamente). Ele denominou essas cadeias de Cadeias Boladas.

Representando a cadeia alimentar como um grafo, todos os pares de espécies (u, v) se relacionam se existe um caminho de u para v OU um caminho de v para u.



Dado uma cadeia alimentar, Bino quer saber se ela é uma Cadeia Bolada ou não.

Entrada

A entrada consiste em múltiplas linhas. A primeira linha contém dois inteiros **N** ($1 \leq$ **N** ≤ 100000) e **M** ($1 \leq$ **M** ≤ 1000000), representando a quantidade de espécies e o número de relações respectivamente. As próximas **M** linhas contêm dois inteiros **U** ($1 \leq$ **U** $\leq$ **N**) e **V** ($1 \leq$ **V** $\leq$ **N**), representando que existe uma relação unidirecional entre **U** e **V**.

Saída

Imprima uma linha com a mensagem "Bolada" (sem aspas) se a cadeia for uma Cadeia Bolada, ou "Nao Bolada" caso contrário.

Exemplos de Entrada	Exemplos de Saída
3 3 1 3 2 3 3 1	Bolada
3 2 1 3 2 3	Nao Bolada

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 fornecidas 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.

<table>
<tr><th>Exemplo de Entrada</th><th>Exemplo de Saída</th></tr>
<tr><td><pre>
4 5
1 4 1
1 3 3
3 4 4
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
</pre></td><td><pre>
Instancia 1
80

Instancia 2
140

Instancia 3
impossivel
</pre></td></tr>
</table>

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** ($1 \le N \le 10^3$), o número de arestas no grafo original **M** ($0 \le M \le 10^5$), o número de arestas a serem inseridas **P** ($0 \le P \le 10^6$) e o número de cores **K** ($1 \le K \le 10^3$). A linha seguinte contém **N** números $\mathbf{X_i}$ indicando a cor do i-ésimo vértice ($1 \le X_i \le K$). As **M** seguintes linhas contém um par de inteiros ($\mathbf{V_i}$, $\mathbf{V_j}$) indicando a presença de uma aresta entre os vértices $\mathbf{V_i}$ e $\mathbf{V_j}$. ($1 \le V_i, V_j \le N$).

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