VIII Maratona de Programação

Universidade de São Paulo

Caderno de Problemas

Departamento de Ciência da Computação, IME–USP

Domingo, 15 de Agosto de 2004

Problema A: Sapos de Tsé-Tsé

`Arquivo: sapos.[c|cpp|java]`

A mosca do sono é uma das pragas mais sérias na China, que causa prejuízos enormes ao governo do país. Populações inteiras de pequenas cidades são picadas pela mosca e acabam caindo no sono durante o trabalho (muitos suspeitam que nem são as moscas as causadoras do problema, mas isso é outra história...).

Preocupados com esta situação os pesquisadores de Engenharia Genética da Universidade de Zhao-Zhao estudaram o genoma de um sapo comedor de insetos da região e descobriram que o padrão de saltos do sapo poderia ser facilmente controlado se uma alteração fosse feita em seu cromossomo 12. Infelizmente nem todos os experimentos resultaram em sucesso e, além de alguns sapos sem pernas e com 12 olhos, os experimentos deram origem a várias espécies de sapos com características diferentes de saltos. O objetivo deste problema é que vocês desenvolvam um programa que, a partir da observação do padrão de saltos de um sapo, verifique se ele é do tipo desejado. Um sapo é do tipo desejado se colocado no canto superior esquerdo de um lago retangular ele cobrir toda a extensão do lago com um número mínimo de saltos.

Para anotar o padrão de saltos de um sapo foram feitos vários experimentos. Em cada experimento o sapo foi colocado em uma posição do lago e se anotou para que posição vizinha ele saltou. As posições vizinhas são ordenadas de 1 a 8 no sentido dos ponteiros do relógio, começando na posição imediatamente acima da posição do sapo, como na figura abaixo.

8	1	2
7	*sapo*	3
6	5	4

Sua tarefa é dada uma instância de um lago, marcado em cada uma de suas posições com o padrão de saltos do sapo, verificar se este, quando colocado no canto superior esquerdo do lago percorre todas as suas posições.

Entrada

São dadas várias instâncias. Cada instância começa com dois inteiros $0 \leq m \leq 1000$ e $0 \leq n \leq 1000$ que definem a dimensão do lago. Em seguida vêm m linhas com n números inteiros, descrevendo o comportamento do sapo quando colocado naquela posição do lago. Valores $m = n = 0$ indicam o final das instâncias e não devem ser processados.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na linha seguinte, você deve imprimir `sim` se o sapo passou por todas as mn posições do lago e `nao` em caso contrário. Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
3 3
3 3 5
5 7 7
3 3 3
2 3
4 4 2
1 1 2
0 0
```

Exemplo de saída

```
Instancia 1
sim

Instancia 2
nao
```

Problema B: Obras da China

`Arquivo: obras.[c|cpp|java]`

Devido à abertura econômica ocorrida na China nos últimos anos, boa parte do país foi transformada em canteiro de obras. Algumas construções em curso são tão monumentais que, juntamente com a já famosa Muralha da China, poderão ser vistas da lua a olho nu.

Uma empreiteira radicada em Shangai é responsável pela execução de várias obras no país. Após algum tempo, os engenheiros da empreiteira perceberam que, a cada nova obra, tinham de resolver um problema semelhante ao que já tinham resolvido no início das obras anteriores. Cansados de realizar sempre os mesmos tipos de cálculos, pediram a sua ajuda na construção de um programa que resolvesse o problema deles, descrito a seguir.

Considere uma obra que tem duração de n semanas. Na i-ésima semana da obra, para $i \leq 1 \leq n$, são necessários f_i funcionários para executá-la. Os custos com recrutamento e instrução de um funcionário são de x *yuan*. Gasta-se y *yuan* para demitir um funcionário. Um funcionário necessário custa z *yuan* por semana e cada funcionário excedente, isto é, cada funcionário contratado que não é necessário em uma semana da obra, custa w *yuan* por semana para a empreiteira. (*yuan* é a moeda chinesa.)

Funcionários podem ser contratados e demitidos a cada semana. Inicialmente, a obra não possui nenhum funcionário. Ao final da mesma, todos os funcionários devem ser demitidos.

O problema consiste em determinar o menor valor possível que a empreiteira deve gastar com funcionários ao longo da obra, satisfazendo sempre as restrições semanais. Ou seja, não pode haver menos de f_i funcionários trabalhando na obra na i-ésima semana.

Entrada

Seu programa deve estar preparado para trabalhar com diversas obras, doravante denominadas instâncias. Cada instância tem a estrutura que segue.

Na primeira linha é fornecido um inteiro n ($0 \leq n \leq 200$) que representa o número de semanas de duração da obra.

Na próxima linha são dados, separados por espaços em branco, n valores inteiros não negativos e menores ou iguais a 50, em que o i-ésimo valor ($1 \leq i \leq n$) representa o número f_i de funcionários necessários na i-ésima semana.

Na linha seguinte, também separados por espaços em branco, são fornecidos quatro inteiros x, y, z e w ($0 \leq x, y, z, w \leq 1000$), em que x é o custo de recrutamento e instrução de um funcionário novo, y é o custo de demitir um funcionário empregado, z é o custo semanal de um funcionário necessário e w é o custo para manter um funcionário excedente, por uma semana, na obra.

Um valor $n = 0$ indica o final das instâncias e não deve ser processado.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na linha seguinte, você deve imprimir o menor valor possível que a empreiteira deve gastar com funcionários ao longo dessa obra.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
5
10 7 9 8 11
80 120 100 160
0
```

Exemplo de saída

```
Instancia 1
7380
```

Problema C: O penúltimo imperador

`Arquivo: imp.[c|cpp|java]`

Muito se conhece do último imperador da China, imortalizado no clássico filme vencedor do Oscar. Porém, seu antecessor, o Imperador Thang Po Lop teve uma vida muito mais interessante, uma vez que morreu ainda na cidade proibida, cercado de concubinas e criados eunucos.

O Imperador Po Lop era um grande colecionador de pauzinhos (daqueles que os orientais utilizam para comer). Desde seus 9 anos ele os guardava e construía com eles enormes labirintos utilizando uma estratégia bastante interessante. Inicialmente Po Lop escolhia um dos pátios retangulares da cidade proibida para construir o labirinto, e esse labirinto sempre ocupava todo o espaço do pátio escolhido. Os pauzinhos eram então colocados nesse pátio aparentemente em lugares aleatórios, sempre paralelos a um dos cantos do pátio. O imperador nunca colocava pauzinhos sobrepostos (nem mesmo parte deles), apesar de ser possível existir cruzamentos ou até mesmo pauzinhos se encostando. Consta na biografia do imperador Po Lop que ele construiu labirintos gigantescos, sempre tomando esses cuidados.

Infelizmente havia um problema. Apesar de exímio construtor de labirintos, o imperador era incapaz de saber se afinal o labirinto continha ou não um caminho ligando sua entrada à sua saída (sempre em lados opostos do pátio). Para saber isso, ele se utilizava de seus eunucos. Ele instruía o eunuco a procurar o caminho naquele labirinto. Muitas vezes, o eunuco dizia não ser possível. O imperador Po Lop se zangava e degolava o infeliz, pois duvidava da resposta do criado. Felizmente, além de muito paciente (não com eunucos) o imperador era bastante cuidadoso, e anotava criteriosamente as informações sobre os labirintos que construía. Estas anotações foram encontradas na biblioteca da cidade proibida quando da revolução e salvas da destruição. Sua tarefa neste problema é resolver finalmente o enigma, verificando se os labirintos construídos pelo Imperador Po Lop têm ou não saída.

Entrada

Seu programa deve estar preparado para trabalhar com diversos labirintos, doravante denominados instâncias. Cada instância é iniciada com uma linha contendo 5 números, ditos $n\ x_i\ y_i\ x_f\ y_f$. O valor n indica o número de pauzinhos que foram usados para construir o labirinto. O par (x_i, y_i) é o canto inferior esquerdo do pátio e também o ponto de partida. O par (x_f, y_f) é o canto superior direito e também ponto de chegada do labirinto. Nas próximas n linhas são dadas as coordenadas $x_1\ y_1\ x_2\ y_2$ representando os extremos (x_1, y_1) e (x_2, y_2) de um dos pauzinhos usados na construção do labirinto. O arquivo de entrada termina com $n < 0$. Pode-se assumir que todos os números dados são inteiros e que $n \leq 1000$.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na linha seguinte, voce deve imprimir `sim` se existir uma maneira de ir do ponto de partida do labirinto até seu ponto de chegada (sem atravessar nenhum pauzinho...), e imprimir `nao` em caso contrário.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
1 0 0 2 2
0 1 1 1
2 0 0 2 2
1 2 1 0
0 1 2 1
-1
```

Exemplo de saída

```
Instancia 1
sim

Instancia 2
nao
```

Problema D: Miai chinês

`Arquivo: miai.[c|cpp|java]`

São notórios os problemas que a China vem enfrentando ao longo do tempo para controlar a explosão populacional que aflige o país. Para piorar a situação, no interior, os casamentos costumam ser tradicionalmente arranjados nas famílias, aumentando as chances do nascimento de crianças com pais aparentados.

Ciente do problema, o governo chinês resolveu criar uma agência oficial de matrimônios. Esta agência deve receber as informações dos jovens que pretendem se casar e decidir se é possível realizar casamentos entre eles que evitem uniões de parentes e de tal forma que nenhum dos jovens termine solteiro. Como em muitos outros países do mundo, na China são permitidos apenas casamentos monogâmicos entre rapazes e garotas.

Sua tarefa neste problema é auxiliar o governo, escrevendo um programa para descobrir se é possível realizar casamentos em dados grupos de jovens.

Entrada

Seu programa deve estar preparado para trabalhar com diversos grupos de jovens, doravente denominados instâncias. Cada instância tem a estrutura que segue.

Na primeira linha são fornecidos inteiros n ($0 \leq n \leq 100$), que representa o número de rapazes e garotas, e m ($0 \leq m \leq 1000$) que representa o número de parentescos existentes entre eles. Não foram incluídos em m parentescos entre um mesmo sexo, pois isso é irrelevante ao problema.

Nas próximas m linhas são fornecidos m pares de números entre 1 e n, inclusive, um par por linha. O primeiro número representa um rapaz e o segundo uma garota que são parentes.

Valores $n = m = 0$ indicam o final das instâncias e não devem ser processados.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na linha seguinte, você deve imprimir `possivel` se for possível realizar os casamentos entre os n rapazes e as n garotas sem que parentes se casem, e imprimir `impossivel` em caso contrário.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
5 5
1 1
2 2
3 3
4 4
5 5
3 5
1 1
1 2
2 2
2 3
3 3
0 0
```

Exemplo de saída

```
Instancia 1
possivel

Instancia 2
possivel
```

Problema E: O país das bicicletas

`Arquivo: bikes.[c|cpp|java]`

Como você já deve saber, a bicicleta é um dos meios de transportes mais populares da China. Quase todos os chineses possuem a sua, e utilizam-na para trabalho, recreação, e outras atividades.

Após muitos anos pedalando, Mr. Lee já não têm a mesma disposição para encarar as diversas subidas da cidade onde mora. E a cidade em que Mr. Lee vive é extremamente montanhosa. Por razões sentimentais, ele não quer mudar para uma cidade mais plana. Resolveu, então, que tentaria evitar grandes altitudes em seus caminhos mesmo que, para isso, tivesse que pedalar um pouco mais.

Mr. Lee obteve com o serviço topográfico chinês uma coleção de mapas de sua cidade, em que cada rua desses mapas possui a informação da maior altitude encontrada quando trafegada. Tudo que ele precisa fazer agora é determinar rotas que minimizem a altura percorrida entre pares (origem, destino).

Sabendo que você planeja visitar a cidade em que ele mora no próximo ano, Mr. Lee pediu sua ajuda. Em uma primeira etapa, ele deseja que você implemente um programa que receba mapas topográficos da cidade e uma coleção de pares (origem, destino). Para cada par, seu programa deve imprimir a maior altura encontrada em uma rota entre a origem e o destino. Lembre-se que as rotas devem minimizar tais alturas. As rotas propriamente ditas, serão determinadas em uma segunda etapa (quando você chegar à China para visitá-lo).

Como o transporte utilizado é uma bicicleta, você pode considerar que todas as ruas da cidade são de mão dupla. Não demore, pois Mr. Lee conta com você. :-)

Entrada

Seu programa deve estar preparado para trabalhar com diversos mapas, doravante denominados instâncias. Cada instância tem a estrutura que segue.

Na primeira linha são fornecidos dois inteiros n ($0 \leq n \leq 100$) e m ($0 \leq m \leq 4950$) que representam, respectivamente, os números de interseções e de ruas. Por razões de clareza, as interseções são numeradas de 1 a n, inclusive; toda rua começa e termina em uma interseção; e não existem interseções fora das extremidades de uma rua.

Nas próximas m linhas são fornecidos três inteiros: i e j ($1 \leq i, j \leq n$) que indicam a existência de uma rua entre as interseções i e j; e h que representa a maior altitude encontrada quando a rua é trafegada. Esses inteiros estão separados por espaços em branco.

Na linha seguinte, é dado um inteiro k ($1 \leq k \leq 50$) que representa o número de pares (origem, destino) que serão especificados nas próximas k linhas. Cada par é formado por dois inteiros i e j como acima. Isto é, origem e destino são interseções de ruas, e também estão separados por espaços em branco.

Valores $n = m = 0$ indicam o final das instâncias e não devem ser processados.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Nas próximas k linhas, você deve imprimir as maiores alturas encontradas nas rotas entre os k pares (origem, destino) fornecidos, um valor por linha, na ordem da entrada.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
12 17
1  4  4
4  7  6
7  10 6
2  5  4
5  8  5
8  11 2
3  6  5
6  9  3
9  12 1
1  2  1
2  3  9
4  5  3
5  6  7
7  8  7
8  9  2
10 11 1
11 12 2
4
1  5
6  8
6  7
11 10
0 0
```

Exemplo de saída

```
Instancia 1
4
3
6
1
```

Problema F: Estimando a produção

`Arquivo: prod.[c|cpp|java]`

A China é uma grande produtora de alimentos, mas também uma enorme consumidora. Pesquisadores chineses perceberam que em certos momentos de sua história a produção agrícola foi maior que o consumo, e em outros momentos esse quadro se inverteu. Preocupados com o futuro da grande nação, passaram a coletar dados sobre a área de plantio, a quantidade de trabalhadores e a produção agrícola. Para melhor investir seus recursos, eles agora desejam fazer uma previsão sobre a produção do país.

O conjunto de dados que os pesquisadores conseguiram coletar é formado por triplas (x_i, y_i, z_i), em que x_i representa a área de plantio, y_i a quantidade de trabalhadores e z_i a produção agrícola. Como essa produção esta ligada diretamente com os demais dados coletados, eles decidiram estimar a produção futura usando a função linear $a_1 + a_2x + a_3y$, que minimiza a soma dos erros quadrados

$$\sum_{i=1}^{n}(z_i - (a_1 + a_2x_i + a_3y_i))^2$$

em que n é o total de triplas disponíveis. Desta forma eles serão capazes de planejar melhor a produção e o consumo dos próximos anos. Seu objetivo é calcular a função linear desejada.

Entrada

A entrada é composta de diversas instâncias. Para cada instância da entrada é dado um número inteiro $3 \leq n \leq 1000$ indicando quantas triplas foram obtidas na coleta de dados. Em cada uma das próximas n linhas é dada uma tripla x_i y_i z_i com a área de plantio (em milhares de hectares), a quantidade de trabalhadores envolvidos (dado em milhares de pessoas), e a produção agrícola (dada em toneladas de alimentos), respectivamente. O arquivo de entrada termina quando for encontrado $n = 0$. Assuma que não existe uma relação linear entre a quantidade de trabalhadores e a área de plantio, ou seja, não existem constantes α, β tais que, para todo i, $x_i = \alpha y_i + \beta$. Assuma também que $0 \leq x_i, y_i, z_i \leq 1000$ e que todos os valores dados são inteiros.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na próxima linha, você deve imprimir os três números a_1 a_2 a_3, representando os coeficientes da função linear procurada. Esses números devem estar truncados em três casas decimais.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
3 3
1 0 0
5 1 1
3 2 2
5
1 3 2
3 7 3
5 10 4
7 400 5
9 4 6
0
```

Exemplo de saída

```
Instancia 1
0.000 0.000 1.000

Instancia 2
1.500 0.500 0.000
```

Problema G: Arqueólogos de Tsing Ling

`Arquivo: tsing.[c|cpp|java]`

A região de Tsing Ling é internacionalmente famosa por grandes achados arqueológicos. Alguns desses achados, como as ruínas de Tsé Lung Zhao, desafiam os melhores cientistas do mundo. Encravados nessas ruínas foram encontrados vários pares de seqüências de caracteres de significados desconhecidos, e tal fato tornou-se o maior enigma de Tsing Ling. Cada par tem uma seqüência menor e uma maior. Acima do portal do mosteiro de Tsing Ling está escrito que aquele que descobrir o significado das seqüências terá a resposta para todas as perguntas do universo.

O arqueólogo alemão Harry Thanan Gruber acha que descobriu o enigma, mas para isso precisa da ajuda de vocês neste problema. Ele acha que a solução do enigma dos caracteres de Tsé Lung Zhao fornecerão os números para decifrar o livro de Tsin Wu, achado na mesma região pelo seu orientador, Dr. Cauchy-Schwartz, no fim do século XIX. Segundo a conjectura de Herr Gruber, a seqüência de números que deve ser usada para decifrar o enigma é dada pelo número de vezes em que cada seqüência menor ocorre na maior como subseqüência.

Entrada

São dados vários pares de seqüências. Para cada par é dado o número $0 \leq m \leq 100$ de caracteres na seqüência menor ou igual e o número $0 \leq n \leq 1000$ de caracteres na seqüência maior. Nas linhas seguintes são dadas as duas seqüências, uma por linha, primeiro a menor e depois a maior. Valores $m = n = 0$ indicam o final dos dados.

Saída

Para cada par de seqüências, doravente denominadas instância, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na próxima linha, você deve imprimir o número de vezes que a seqüência menor ocorre como subseqüência da maior. Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
3 10
*x2
a**x***xX2
0 0
```

Exemplo de saída

```
Instancia 1
7
```

Problema H: Grupos da Universidade de Sing Pil

`Arquivo: grupos.[c|cpp|java]`

Na famosa Universidade de Sing Pil os estudantes sempre fazem os trabalhos em grupo. As regras para a formação dos grupos, no entanto, são estritas e o reitor sempre verifica se não há um grupo de estudantes que violou as regras.

Bom, para falar a verdade, a única regra existente remonta à criação da universidade. Naquela época os alunos compunham grupos de três alunos para fazer as tarefas. Quatro alunos, chamados Ting, Ling, Xing e Ming eram muito amigos e, para todas as tarefas que precisavam ser feitas, montavam um grupo entre eles. Isso era muito ruim, porque obrigar as tarefas em grupos visava a aumentar a interação entre os alunos. Desde então proibiu-se em Sing Pil a formação de **quadrados**, isto é, que quatro alunos montem quatro grupos em que apenas eles são os membros.

No caso dos alunos $\{Ting, Xing, Ling, Ming\}$ (usaremos apenas a primeira letra para simplificar), um quadrado seria formado pelos quatro grupos a seguir: $\{TLX, TXM, MXL, LMT\}$.

Sua tarefa neste problema é escrever um programa para ajudar o reitor da universidade a verificar se existem ou não quadrados nos grupos.

Entrada

São dadas várias instâncias. Para cada instância é dado o número $0 \leq m \leq 50$ de grupos. O valor $m = 0$ indica o fim dos dados e não deve ser processado.

Cada estudante em Sing Pil é identificado com um número inteiro entre 1 e 100, inclusive. Nas próximas m linhas são dados, em cada linha, três números correspondentes a três estudantes que formam um grupo.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Caso não existam quadrados nos grupos, seu programa deve imprimir `ok`. Em caso contrário, seu programa deve imprimr todos os quadrados encontrados, um por linha, com os números dos alunos separados por um espaço em branco. Para facilitar a leitura do reitor, os números dos alunos em um quadrado deverão estar em ordem crescente e os quadrados deverão estar listados em ordem lexicografica crescente.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
5
1 2 3
4 5 6
1 5 6
3 6 7
8 6 1
8
1 2 3
4 5 6
1 5 4
7 6 4
5 1 6
3 6 5
6 1 4
2 5 6
0
```

Exemplo de saída

```
Instancia 1
ok

Instancia 2
1 4 5 6
```

Problema I: Centro de convenções

`Arquivo: centro.[c|cpp|java]`

Já com a Final Mundial da Maratona de Programação em mente, o governo chinês iniciou um projeto para a construção de um centro de convenções novo. Esse centro será o mais moderno do mundo, com toda a infra-estrutura para sediar importantes eventos. O governo já decidiu (e se decidiu está decidido) construí-lo no formato de uma circunferência. Quando visto de cima esse novo centro, com auxílio de toda sua iluminação de última geração, irá dar a impressão de ser uma grande nave espacial redonda. Com truques de luzes, pretende-se ainda criar a impressão de movimento para o imponente prédio.

Porém todos sabem que a China possui um grande problema de espaço físico, e o único lugar disponível para a construção fica nos arredores de uma antiga floresta de árvores milenares. Para deixar o projeto ainda mais atraente, decidiu-se que o centro será construído dentro da floresta, mas sem derrubar uma única árvore. A sorte do projetista é que a floresta é esparsa, e existe bastante espaço entre as árvores em alguns lugares. Como se deseja criar o maior (no sentido da área construída) centro de convenções possível, sua tarefa é ajudar a encontrar o melhor lugar para a construção. Seu objetivo é encontrar as coordenadas do ponto central da construção, que deve estar dentro do fecho convexo induzido pelas árvores.

Entrada

A entrada é composta de diversas instâncias. Cada instância inicia-se com uma linha contendo o número $0 \leq n \leq 1000$ de árvores da floresta, seguida por n linhas contendo os pares ordenados x_i y_i, que representam as coordenadas das árvores da floresta. Todas as coordenadas dadas são inteiras. A entrada termina com $n = 0$.

Saída

Para cada instância solucionada, você deverá imprimir um identificador `Instancia h` em que `h` é um número inteiro, seqüencial e crescente a partir de 1. Na linha seguinte você deve imprimir a posição x y ideal para o ponto central do centro de convenções. Caso exista mais que um ponto ideial para a construção, imprima aquele com o menor valor para x. Caso ainda exista mais que uma opção, imprima aquele com o menor valor para y. Trunque os números impressos em exatamente três casas decimais. Caso não seja possivel construir o centro, escreva a palavra `impossivel` na linha.

Uma linha em branco deve separar a saída de cada instância.

Exemplo de entrada

```
4
0 0
2 2
0 2
2 0
3
0 0
10 10
6 4
5
0 0
3 3
1 1
3 1
0 3
0
```

Exemplo de saída

```
Instancia 1
1.000 1.000

Instancia 2
impossivel

Instancia 3
1.500 2.500
```