www.ufes.br/topcom3

Topcom 3

Torneio de Programação de Computadores

CADERNO DE QUESTÕES

Organização

Apoio

Patrocínio

Fundo de Apoio à Ciência e Tecnologia do Município de Vitória

Problema A

Loteria Flexível

Introdução

Uma população não satisfeita com os jogos padronizados da loteria fornecidos pelo governo local decidiu criar a sua própria loteria de forma bastante flexível e configurável, pois assim, argumentavam os cidadãos, o jogo ficaria mais emocionante e atrairia mais jogadores. As informações de uma rodada no jogo são: uma seqüência de números sorteada, várias seqüências de números que representam as apostas e as categorias de acertos que classificam a aposta de acordo com a quantidade de números corretos. A flexibilidade desta loteria é norteada pela possibilidade de configurar a quantidade de números sorteados, a quantidade de números apostados, as categorias e a faixa de números sorteados e apostados.

Entrada

A primeira linha deverá conter a quantidade de números sorteados, a faixa permitida (composta por valor inferior e valor superior) e os números sorteados, sempre em ordem crescente.
Exemplo de primeira linha: “04 01 10 04 06 09 10” = 4 números sorteados de 1 a 10, são eles: 04, 06, 09 e 10.
A segunda linha contém a quantidade de categorias seguida das próprias categorias que classificam as apostas ganhadoras.
Exemplo de segunda linha: “03 00 03 04” = 3 categorias onde as apostas ganhadoras deverão ter 0 (nenhum), 3 ou 4 números iguais aos sorteados.
A terceira linha contém a quantidade de apostas.
A quarta linha em diante contém as apostas, cada uma representada pela quantidade de números apostados seguida dos números apostados.
Admita como limites máximos:

- 50 números sorteados
- 51 categorias

É admitido previamente que:

- Os números sorteados estão dentro da faixa permitida para números sorteados.
- O número que representa a categoria é no mínimo 00 (zero) e no máximo a quantidade de números sorteados.

Saída

O programa deverá gerar um inteiro representando cada categoria com seus respectivos números de apostas ganhadoras. A formatação dos dados de saída deve ser:

(nº da categoria) (espaço simples) (nº de ganhadores da categoria) (\n)

Exemplo de Entrada
04 01 10 04 06 09 10 03 00 03 04 05 04 03 06 09 10 06 01 02 03 06 09 10 04 05 07 09 10 07 01 04 05 06 07 09 10 04 02 03 05 08

Exemplo de Saída
0 1 3 2 4 1

Problema B

MyDoom.Z

Introdução

A Internet mundial sofre todos os dias com novas versões do vírus MyDoom. A última das versões, a MyDoom.Z, objetiva aumentar o fluxo de dados nas redes e assim freá-las. Para isso, depois de passar por um roteador, ele faz um “append” no seu código, ou seja, ele acrescenta seu código original após o código atual.

A empresa alemã “Der Virusjäger” quer saber por quantos roteadores um determinado vírus passou. Você foi contratado para essa tarefa.

Entrada

Em cada linha da entrada você terá o código atual do vírus. O código é uma seqüência com N caracteres alfa-numéricos ($1 \leq N \leq 1.000.000$). Uma linha contendo um único ponto indica a terminação dos testes.

Saída

Para cada teste você deve imprimir o número de roteadores pelos quais o vírus passou.

Exemplo de Entrada
Abcd aaaa abcabcabcabc hLp15hLp15Z 101101 .

Exemplo de Saída
1 4 4 1 2

Problema C

Fusão de Redes

Introdução

Existe uma onda de fusões e aquisições de empresas de informática. Seguindo esta onda, a empresa BigSoft Ltda adquiriu a SmallSoft S/A.

A BigSoft possui uma rede de comunicação de alta velocidade conectando todas as cidades onde as duas empresas atuam, já a SmallSoft possui uma rede bem menor, de modo que poucas cidades estão conectadas. A direção da BigSoft resolveu otimizar a rede de comunicação da empresa utilizando os poucos elos de comunicação da SmallSoft, caso isto seja conveniente economicamente.

Você deve implementar um programa que modifique a rede de comunicação da BigSoft de forma ótima.

O custo de manutenção de uma rede é a soma dos custos de manutenção dos elos que formam a rede. Note que as empresas originais BigSoft e SmallSoft atuam nas mesmas cidades, não existem ciclos na rede e os custos são inteiros e variam entre 0 e 10000. As cidades são numeradas seqüencialmente a partir de 1 e o tamanho máximo da rede é de 1000 cidades.

Entrada

A entrada consiste de vários casos de teste separados por um caractere cifrão ($). Cada caso de teste consiste das redes de comunicação da BigSoft e da SmallSoft. Cada linha da entrada representa um link de comunicação (as duas cidades envolvidas) e seu respectivo custo de manutenção. Por exemplo, se existir um link entre a cidade 1 e a cidade 2 com custo de manutenção de R$ 1700,00, este será representado por: 1 2 1700. A rede da BigSoft é separada da rede da SmallSoft por um ponto (.), os links da BigSoft aparecem primeiro e os casos de teste terminam com um asterisco (*).

Saída

A saída consiste dos elos substituídos e dos novos elos colocados no lugar, de modo que se o elo 1 2 1700 é substituído pelo elo 1 4 100, a saída deverá conter: 1 2 1700 -> 1 4 100. Os casos são separados por “$“.

Exemplo de Entrada
1 2 400 2 3 700 3 9 200 3 7 400 3 4 700 7 6 200 6 8 100 7 5 300 . 3 9 100 8 2 800 1 5 500 $ 7 2 100 2 1 200 2 3 200 3 5 400 5 4 300 3 6 300 . 2 5 100 *

Exemplo de Saída
3 9 200 -> 3 9 100 2 3 700 -> 1 5 500 $ 3 5 400 -> 2 5 100 $

Problema D

Jimmy Neutron

Introdução

Jimmy Neutron quer fazer um curso de graduação em cada universidade. Ele quer gastar o menor número possível de períodos para concluir cada curso. Como é um gênio, ele pode fazer qualquer número de disciplinas em um mesmo período, só não pode fazer todas as disciplinas de uma vez, pois deve atender aos pré-requisitos estabelecidos na grade curricular.

Ele precisa que você faça um programa que forneça um número mínimo de períodos que este terá que fazer, supondo que a única restrição que ele encontra é o correto cumprimento dos pré-requisitos.

Entrada

A entrada é composta por vários conjuntos de teste que formam as grades dos cursos de cada universidade que ele irá cursar.

Cada conjunto de teste possui os seguintes dados:

Na primeira linha: um número N ($1 \leq N \leq 1000$) que representa o número total de disciplinas que o curso possui.

Nas N linhas subseqüentes: O código C ($1 \leq C \leq 1000$) de uma dada disciplina seguido de uma lista de códigos de disciplinas que são seus pré-requisitos. A lista de códigos de disciplinas é terminada por –2.

Quando não houver pré-requisito o código aparece sozinho na linha seguido de -2. O algarismo –1 indica o término do conjunto de testes.

Saída

Para cada conjunto de teste seu programa deve produzir o número mínimo de períodos que Jimmy deverá fazer para concluir o curso. A primeira linha de um conjunto de saída deve conter o identificador do conjunto, no formato “Teste n”, onde n é numerado a partir de 1.

Exemplo de entrada	Exemplo de saída
8 10 –2 12 –2 27 12 10 –2 32 24 27 –2 24 13 –2 13 12 –2 47 24 –2 40 –2 5 50 10 20 30 40 –2 10 –2 20 –2 30 –2 40 20 –2 -1	Teste 1 4 Teste 2 3

Problema E

Picking em Armazéns

Introdução

O problema de *picking* em armazéns é uma das aplicações tradicionais de algoritmos de otimização combinatória em logística. Nele, são conhecidos os seguintes dados:

- As dimensões de um armazém;
- A distribuição das estantes em um armazém;
- A localização (no armazém) onde os atendentes pegam o pedido dos clientes;
- A localização (no armazém) onde os atendentes entregam a caixa com os produtos pedidos, após coletá-los (*picking*) nas estantes;
- A localização dos estoques de produtos nas estantes;
- Os pedidos feitos por cada cliente, onde constam diversos produtos.

O objetivo do problema é obter a ordem de coleta dos produtos, de maneira que o caminho que o atendente irá seguir entre as estantes, pegando os produtos e depois voltando ao ponto "zero", seja o de menor comprimento (maximizando, assim, a sua produtividade).

A figura a seguir ajuda a entender melhor o problema. Nele, o atendente tem que recolher os produtos A-F, e voltar ao ponto "zero".

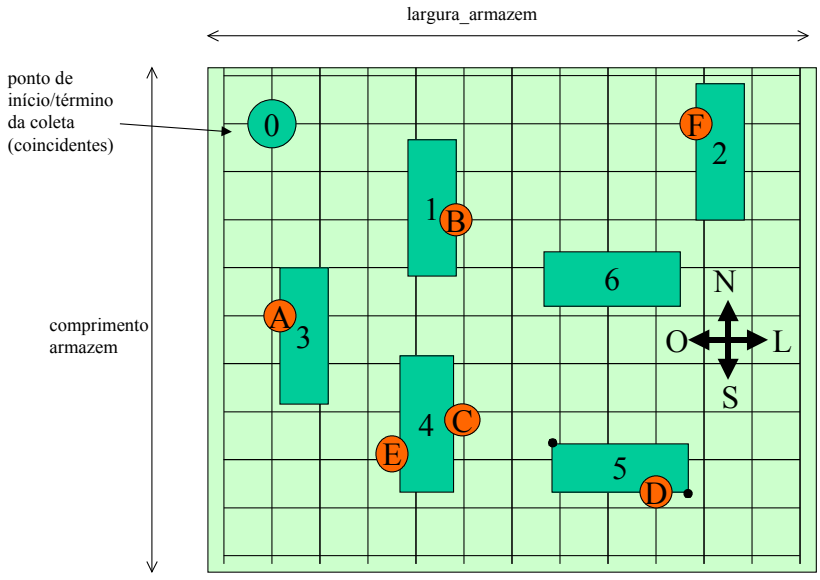

Figura 1

Observe que:

 São distribuídos "E" estantes e "P" produtos pelo armazém;
 O sistema de coordenadas que será utilizado para orientar os atendentes tem como origem (coordenadas = (0;0)) o ponto superior-esquerdo do armazém. A localização de um ponto é dada por um par (X;Y), onde X é a dist,ncia leste-oeste em relação ‡ origem e Y é a dist,ncia norte-sul;
 O ponto onde os atendentes pegam os pedidos e entregam os produtos é localizado de acordo com este sistema de coordenadas. No exemplo da figura 1, a localização deste ponto "zero" é (1 ; 1);
 Cada estante retangular é localizada espacialmente pela indicação da posição dos seus quatro cantos, na seguinte ordem: superior-esquerdo, superior-direito, inferior-esquerdo e inferior-direito. Na figura 1, a localização dos cantos da estante 1 é:

Canto superior esquerdo= (3.8 ; 1.4);
Canto superior direito = (4.8 ; 1.4);
Canto inferior esquerdo = (3.8 ; 4.2);
Canto inferior direito = (4.8 ; 4.2).

 Os produtos são localizados a partir das suas coordenadas. Por exemplo, o produto B está localizado na posição (X;Y) = (4.8; 3.0). Um produto esta sempre localizado nos limites de uma estante.

Primeiramente deve-se montar uma matriz de dist,ncias que tem como nós o ponto “zero”, os cantos de cada estante e os produtos. Cada posição da matriz irá conter a dist,ncia em linha reta entre todos estes pontos, tomados dois-a-dois (par origem-destino). Contudo, sempre que não for possível caminhar em linha reta de um ponto a outro ("atravessando" a área de uma estante), deve-se atribuir ‡quele par origem-destino um comprimento "infinito" (para o problema em questão, considere 999.000000 como o valor para “infinito”).

Sua tarefa agora é gerar uma matriz com N linhas por N colunas, com o comprimento do menor caminho (respeitando as restrições impostas no enunciado) entre cada par origem-destino no grafo (matriz) de entrada.

Entrada

Na primeira linha, um número inteiro N, representando o número de nós do "grafo".
Na segunda linha, um número inteiro P, representando o número de nós que contém produtos.
Uma matriz com 4*E+P+1 linhas por 4*E+P+1 colunas descrevendo as

dist,ncias entre cada par origem-destino dos pontos citados acima. Para separar uma entrada da outra, use o caractere “ * “ (asterisco). Para indicar o fim dos testes, use o caractere “.” (ponto).

Saída

Soma das células de uma matriz com N linhas por N colunas, com o comprimento do menor caminho entre cada par origem-destino no grafo de entrada. Para separar uma saída da outra, use o caractere “ * “ (asterisco). Para indicar o fim do programa, insira “.” (ponto).

Observando a Figura 1, não esqueça que a menor dist,ncia entre o ponto “zero” e o produto F é uma linha reta. Entre o ponto “zero” e o produto B, siga em linha reta até o canto superior-direito da estante 1, e depois em linha reta até o produto para encontrá-lo percorrendo o menor caminho.

Exemplo de Entrada	Exemplo de Saída
6.000000 1.000000 0.000000 1.389244 2.080865 3.275668 999.000000 999.000000 1.389244 0.000000 1.000000 2.000000 999.000000 999.000000 2.080865 1.000000 0.000000 999.000000 2.000000 999.000000 3.275668 2.000000 999.000000 0.000000 1.000000 0.900000 999.000000 999.000000 2.000000 1.000000 0.000000 0.100000 999.000000 999.000000 999.000000 0.900000 0.100000 0.000000 * 6.000000 1.000000 0.000000 1.389244 2.080865 3.275668 999.000000 999.000000 1.389244 0.000000 1.000000 2.000000 999.000000 999.000000 2.080865 1.000000 0.000000 999.000000 2.000000 999.000000 3.275668 2.000000 999.000000 0.000000 1.000000 0.900000 999.000000 999.000000 2.000000 1.000000 0.000000 0.100000 999.000000 999.000000 999.000000 0.900000 0.100000 0.000000 .	66.005 * 66.005 .

Problema F

Picking em Armazéns

Introdução

Seguindo as especificações definidas no Problema E, sua tarefa agora é determinar o comprimento do roteiro de *picking* ótimo em um grafo, iniciando no ponto “zero”, passando obrigatoriamente pelos nós que "contém produtos", mas podendo passar por nós intermediários "sem produtos" para encurtar caminho. Após recolher todos os produtos, você deve voltar ao ponto inicial.

Entrada

Um número inteiro N, representando o número de nós do grafo.
Um número inteiro P, representando o número de nós que contém produtos.
Uma matriz contendo os comprimentos dos caminhos mínimos entre cada par de nós do grafo**.** Um exemplo deste tipo de matriz é a matriz gerada pelo problema E, que apresenta a matriz de dist,ncias entre N=6 nós, sendo que os P=1 últimos nós contém produtos.
Para separar uma entrada da outra, use o caractere “ * “ (asterisco). Para indicar o fim dos testes, use o caractere “.” (ponto).

Saída

O comprimento do roteiro de *picking* ótimo.
Para separar uma saída da outra, use o caractere “ * “ (asterisco). Para indicar o fim do programa, insira “.” (ponto).

Exemplo de Entrada
6.000000 1.000000 0.000000 1.389244 2.080865 3.275668 4.080865 4.175668 1.389244 0.000000 1.000000 2.000000 3.000000 2.900000 2.080865 1.000000 0.000000 3.000000 2.000000 2.100000 3.275668 2.000000 3.000000 0.000000 1.000000 0.900000 4.080865 3.000000 2.000000 1.000000 0.000000 0.100000 4.175668 2.900000 2.100000 0.900000 0.100000 0.000000 * 6.000000 1.000000 0.000000 1.389244 2.080865 3.275668 4.080865 4.175668 1.389244 0.000000 1.000000 2.000000 3.000000 2.900000 2.080865 1.000000 0.000000 3.000000 2.000000 2.100000 3.275668 2.000000 3.000000 0.000000 1.000000 0.900000 4.080865 3.000000 2.000000 1.000000 0.000000 0.100000 4.175668 2.900000 2.100000 0.900000 0.100000 0.000000 .

Exemplo de Saída
8.351 * 8.351 .