

[www.ufes.br/topcom3](http://www.ufes.br/topcom3)

# Topcom 3

## Torneio de Programação de Computadores

# AQUECIMENTO

### Organização



### Apoio



### Patrocínio



# Problema A

## Agenda do reitor

### Introdução

O reitor de uma universidade é uma pessoa muito ocupada, tendo que participar de reuniões e seminários com frequência. Quando surge a necessidade de participar de um compromisso é necessário verificar as possíveis disponibilidades em sua agenda para que ele possa escolher qual período é mais adequado para atender a essa demanda.

A agenda do reitor é composta por uma seqüência de períodos nos quais ele pode ou não agendar um compromisso ou parte dele. Se o período está marcado (isto é, representado por 1) significa que aquele período já foi agendado para um compromisso e se encontra indisponível. Se o período não está marcado (isto é, representado por zero na agenda), ele está disponível para agendamento.

Os compromissos do reitor podem demandar uma seqüência contígua ou intercalada de períodos. Uma seqüência intercalada é indicadora de que nos períodos entre um período a ser alocado e outro pode haver ou não um outro compromisso já alocado. O caractere ? é usado para representar esses períodos na especificação da seqüência intercalada de um compromisso a ser agendado.

Sua tarefa é escrever um programa que, dada uma seqüência de períodos representando um intervalo do estado atual da agenda e a especificação de uma seqüência de períodos demandada por um compromisso, apresente todas as possíveis alternativas de disponibilidade na agenda do reitor.

Considere que o número máximo de períodos de um intervalo é 1000 (mil) e que o número máximo de períodos contíguos para um compromisso é 100 (cem).

Como exemplo, imagine que um intervalo do estado atual da agenda seja equivalente a uma manhã de trabalho (das 8h00min às 12h00min). Considerando cada hora do dia como sendo um período, agendou-se um compromisso de 8h00min às 9h00min. Na agenda, o horário correspondente estaria preenchido e marcado com 1. Depois de marcado o compromisso, a agenda estaria desta forma: 1 0 0 0. Agendando-se agora um compromisso intercalado que ocupe o período das 9h00min

(9h00min – 10h00min) e o período das 11h00min (11h00min – 12h00min), a agenda estaria assim configurada: 1 1 0 1.

## Entrada

Uma entrada é composta por duas linhas. A primeira contém uma seqüência de períodos representando um intervalo no estado atual da agenda e a segunda linha contém a especificação de uma seqüência de períodos demandada por um compromisso. Observe que, quando se tratar de um compromisso intercalado, os períodos não considerados entre as partes desse compromisso devem ser indicados por ?.

Para indicar o fim de cada linha da entrada, usa-se o caractere \*. Para determinar o fim dos testes, usa-se o caractere #.

## Saída

Seu programa deve produzir vários conjuntos de saída representando as possíveis alternativas de agendamento. Cada conjunto é composto por duas linhas. A primeira linha deve conter um número indicador do período no qual a alternativa de agendamento começa (considere que o primeiro período da agenda é representado pelo índice zero). A segunda linha deve apresentar a seqüência completa de períodos da alternativa de agendamento, tal como se encontra na agenda. Para indicar o fim dos conjuntos de saída para uma dada entrada, use o caractere \$. A grafia mostrada no Exemplo de Saída, a seguir, deve ser observada rigorosamente.

Exemplos de Entradas
1 1 0 1 0 1 0 0 1 0 0 0 0 *
0 ? 0 ? ? 0 *
1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 *
0 ? 0 0 *
1 1 0 0 0 0 1 *
0 0 0 *
#

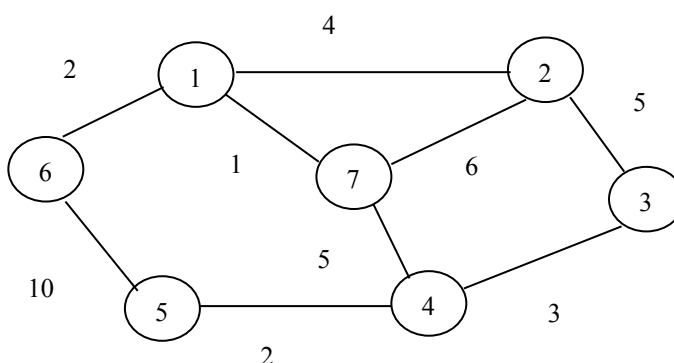
Exemplos de Saídas
2
0 1 0 1 0 0
4
0 1 0 0 1 0
7
0 1 0 0 0 0
\$
9
0 1 0 0
11
0 0 0 0
12
0 0 0 0
\$
2
0 0 0
3
0 0 0
\$

# Problema B

## Rede de comunicação de dados

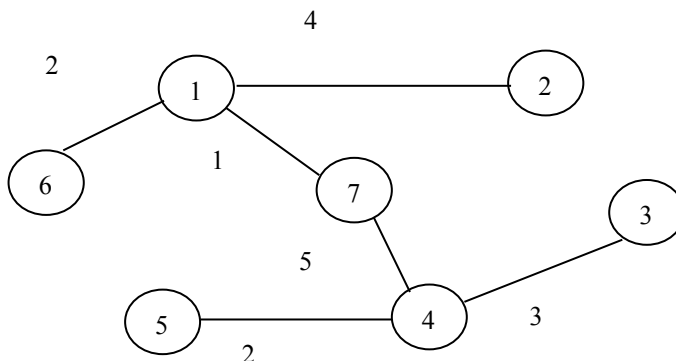
### Introdução

Uma rede de distribuição de dados possui estações das quais os dados são transmitidos. Deve haver conexão entre todas as estações. Não há a exigência de ligação direta entre 2 estações, apenas que exista ligação entre quaisquer pares de estações. Existem custos de manutenção entre as estações.



Contudo, pode haver problemas de comunicação entre algumas estações, mas o sistema tem que ser protegido de tal forma que exista uma, ou mais, “redes de conexão seguras” que garantam a ligação entre todas as estações.

Sua tarefa é encontrar uma “rede de conexão segura” com menor custo de manutenção. Para a rede descrita na figura acima, segue uma das “redes de conexão segura” com menor custo de manutenção possível.



## Entrada

O caso de teste é composto da seguinte maneira:

Um inteiro  $N$  ( $2 \leq N \leq 50$ ) que representa o número de estações. Segunda linha: um inteiro  $M$  ( $1 \leq M \leq 1275$ ) que representa o número de ligações entre as estações;

Terceira linha em diante:  $M$  linhas com pares de estações onde existam ligações e seus respectivos custos de manutenção.

## Saída

Para o teste você deve imprimir o valor da soma dos custos de manutenção.

Exemplo de Entrada
7
9
1 2 4
1 6 2
1 7 1
2 3 5
2 7 6
3 4 3
4 5 2
4 7 5
5 6 10

Exemplo de Saída
17

# Problema C

## *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".

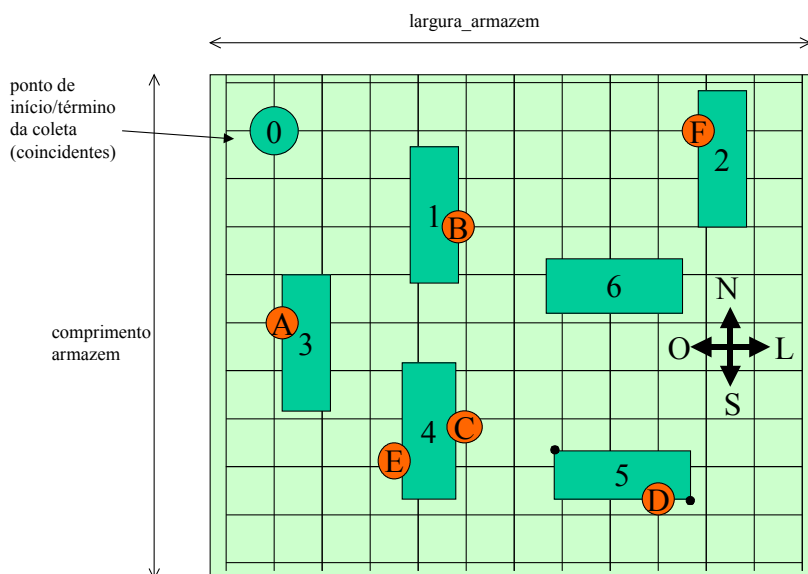


Figura 1

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 distância leste-oeste em relação à origem e Y é a distância 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 está sempre localizado nos limites de uma estante.

Sua tarefa é montar uma matriz de distâncias que tem como nós o ponto “zero”, os cantos de cada estante e os produtos. Cada posição da matriz irá conter a distância 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 àquele par origem-destino um comprimento "infinito" (para o problema em questão, considere 999.000000 como o valor para “infinito”). Seu programa gerará então, como saída, um valor para a soma dos elementos da matriz em questão, com precisão de 3 (três) casas decimais.

## Entrada

Os dados serão passados de acordo com as especificações seguintes:

- Linha 1: o número de estantes ("E") e o número de produtos ("P");
- Linha 2: o par de coordenadas do ponto “zero”;
- Linhas 3 até ((4xE)+2): para cada estante, quatro linhas contendo, respectivamente, os quatro pares de coordenadas. Ou seja, um par em cada linha (separando as coordenadas por um espaço simples), em um total de (8 x "E") valores, seguindo a ordem: superior-esquerdo,



superior-direito, inferior-esquerdo, inferior-direito.

□ Linhas  $((4 \times E)+3)$  até  $((4 \times E)+P+2)$ : para cada produto, uma linha, contendo as coordenadas da localização de cada produto.

Exemplo de Entrada (com um teste apenas):

```
2.000000 4.000000
0.500000 0.500000
0.400000 2.000000
1.400000 2.000000
0.400000 3.000000
1.400000 3.000000
2.500000 0.400000
3.500000 0.400000
2.500000 2.100000
3.500000 2.100000
0.400000 2.800000
3.300000 2.100000
3.500000 0.700000
2.500000 0.800000
```

Observe também que:

1. O armazém e as estantes são sempre retangulares, e são localizadas espacialmente de forma a que se possa caminhar entre elas e a parede sem obstrução. As paredes das estantes são sempre paralelas às paredes do armazém.
2. Os atendentes não conseguem pular sobre ou atravessar as estantes, nem pegar produto de um lado da estante estando em outro lado (isto é, não dá para passar o braço através da estante).
3. O ponto de recepção dos pedidos coincide com o ponto de entrega das mercadorias.
4. Os números máximo e mínimo de estantes são:  $1 \leq E \leq 10$ .
5. Os números máximo e mínimo de produtos são:  $1 \leq P \leq 10$ .
6. As entradas terão 6 casas decimais de precisão.
7. Cada valor da entrada, na mesma linha, estará separado por um espaço em branco.
8. As entradas devem ser separadas por “ \* ” (asterisco). A massa de teste finaliza-se com “.” (ponto final).

## Saída

Soma das células de uma matriz com  $4 \cdot E + P + 1$  linhas por  $4 \cdot E + P + 1$  colunas descrevendo as distâncias entre cada par origem-destino dos pontos citados acima.

Nota: a ordem dos pontos é a mesma ordem da entrada de dados (primeiro o ponto "zero", depois os quatro cantos de cada estante, depois os produtos, na ordem da leitura dos dados).

Exemplo de matriz para a entrada do exemplo anterior:

0,000	1,503	1,749	999,000	999,000	2,002	999,000	2,561	999,000	999,000	999,000	999,000	2,022
1,503	0,000	1,000	1,000	999,000	2,640	999,000	999,000	999,000	0,800	999,000	999,000	2,419
1,749	1,000	0,000	999,000	1,000	1,942	999,000	1,105	999,000	999,000	999,000	999,000	1,628
999,000	1,000	999,000	0,000	1,000	999,000	999,000	999,000	999,000	0,200	999,000	999,000	999,000
999,000	999,000	1,000	1,000	0,000	2,823	999,000	1,421	2,285	999,000	2,102	999,000	2,460
2,002	2,640	1,942	999,000	2,823	0,000	1,000	1,700	999,000	999,000	999,000	999,000	0,400
999,000	999,000	999,000	999,000	999,000	1,000	0,000	999,000	1,700	999,000	999,000	0,300	999,000
2,561	999,000	1,105	999,000	1,421	1,700	999,000	0,000	1,000	999,000	0,800	999,000	1,300
999,000	999,000	999,000	999,000	2,285	999,000	1,700	1,000	0,000	999,000	0,200	1,400	999,000
999,000	0,800	999,000	0,200	999,000	999,000	999,000	999,000	999,000	0,000	999,000	999,000	999,000
999,000	999,000	999,000	999,000	2,102	999,000	999,000	0,800	0,200	999,000	0,000	999,000	999,000
999,000	999,000	999,000	999,000	999,000	999,000	0,300	999,000	1,400	999,000	999,000	0,000	999,000
2,022	2,419	1,628	999,000	2,460	0,400	999,000	1,300	999,000	999,000	999,000	999,000	0,000

Esta matriz pode ser lida da forma abaixo. Acompanhe os valores da primeira linha:

"A distância do ponto "zero" até ele mesmo é zero."

"A distância do ponto "zero" até o canto "superior-esquerdo" da estante-1 é 1.503."

"A distância do ponto "zero" até o canto "superior-direito" da estante-1 é 1.749."

"A distância do ponto "zero" até o canto "inferior-esquerdo" da estante-1 é "infinito"."

"A distância do ponto "zero" até o canto "inferior-direito" da estante-1 é "infinito"."

"A distância do ponto "zero" até o canto "superior-esquerdo" da estante-2 é 2.002."

"A distância do ponto "zero" até o canto "superior-direito" da estante-2 é "infinito"."

"A distância do ponto "zero" até o canto "inferior-esquerdo" da estante-2 é 2.561."

"A distância do ponto "zero" até o canto "inferior-direito" da estante-2 é "infinito"."

"A distância do ponto "zero" até o produto A é "infinito"."

"A distância do ponto "zero" até o produto B é "infinito"."

"A distância do ponto "zero" até o produto C é "infinito"."

"A distância do ponto "zero" até o produto D é 2.022."

O mesmo se repete para a segunda linha, sendo:

"A distância do canto "superior-esquerdo" da estante-1 até o ponto "zero" é 1.503."

"A distância do canto "superior-esquerdo" da estante-1 até ele mesmo é zero."

"A distância do canto "superior-esquerdo" da estante-1 até o canto "superior-direito" da estante-1 é 1.000."

...

Observação:

1. Os resultados devem ser apresentados com arredondamento de 3 casas decimais.

2. Lembre-se de arredondar o valor somente após o resultado da soma.

Exemplo de Entrada	
2.000000	4.000000
0.500000	0.500000
0.400000	2.000000
1.400000	2.000000
0.400000	3.000000
1.400000	3.000000
2.500000	0.400000
3.500000	0.400000
3.500000	0.400000
2.500000	2.100000
3.500000	2.100000
0.400000	2.800000
3.300000	2.100000
3.500000	0.700000
2.500000	0.800000
*	
1.000000	1.000000
0.800000	0.800000
1.500000	2.000000
2.500000	2.000000
1.500000	4.000000
2.500000	4.000000
2.400000	4.000000
.	

Exemplo de Saída
93996.925
*
12015.492
.