OBI2007

Caderno de Tarefas

Modalidade Programação • Seletiva, Teste 2

A PROVA TEM DURAÇÃO DE DUAS HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 2 páginas (não contando esta folha de rosto), numeradas de 1 a 2. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas não estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
  - em Pascal: *readln, read, writeln, write*;
  - em C: *scanf, getchar, printf, putchar*;
  - em C++: as mesmas de C ou os objetos *cout* e *cin*.
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Sociedade Brasileira de Computação
www.sbc.org.br

Estrada Romana

Nome do arquivo fonte: `romama.c`, `romama.cpp`, *ou* `romama.pas`

A civilização romana dominou a Europa do século III A.C. até o século V D.C. e foi responsável por grandes construções como aquedutos, arenas esportivas e estradas. Glaubus era um construtor romano e ficou encarregado de pavimentar um conjunto de estradas já existentes, de tal forma que entre duas cidades do império sempre deve existir um caminho pavimentado entre elas, mesmo que para tal seja necessário passar por outras cidades. Glaubus dispõe de uma única equipe de construção, permitindo que somente um trecho seja pavimentado a cada instante.

Para produzir as lajes necessárias para a pavimentação foram contratadas P pedreiras e cada entrega as lajes com um determinado comprimento T em metros e a largura padrão de uma estrada romana. Duas pedreiras distintas não entregam pedras com o mesmo comprimento. Pode-se utilizar qualquer combinação de possível de pedras para formar um determinado trecho de estrada, entretanto, é proibido cortar as pedras. Por exemplo, se existem pedras de 1, 2, 3 e 4 metros, podemos pavimentar um trecho de estrada de 5 metros de seis formas distintas: 5 pedras de comprimento 1; 1 pedra de comprimento 1 e duas de comprimento 2; 3 pedras de comprimento 1 e uma de comprimento 2; 2 pedras de comprimento 1 e uma de comprimento 3; e uma pedra de comprimento 2 e uma de comprimento 3. Nenhuma outra combinação além dessas duas é possível para o trecho de 5 metros com essas pedras.

Note que, no exemplo anterior, se retirarmos as pedras de comprimento 1, passamos a ter somente uma combinação para o trecho de 5 metros: uma pedra de comprimento 2 e uma pedra de comprimento 3. Além disso, nesse caso não é mais possível pavimentar um trecho de comprimento 1.

Os conselheiros do império notaram que o tempo de pavimentação de cada trecho é proporcional ao número de combinações possíveis de pedras para se pavimentar esse trecho, exceto se houver zero combinações possíveis, pois nesse caso o trecho não pode ser pavimentado. Glaubus então deseja pavimentar o conjunto de estradas que ligue todas as cidades e cujo tempo total de pavimentação seja o menor possível, supondo que os trechos sejam pavimentados em seqüência, um imediatamente após o outro. Como Glaubus vive no sec. II D.C mas possui poderes místicos, ele entrou em contato com você no futuro e pediu sua ajuda.

Tarefa

Escreva um programa que, dado o número de cidades romanas, o número de pedreiras, os possíveis trechos a serem pavimentados e suas distâncias, determine o menor tempo possível para se pavimentar os trechos que ligam as cidades romanas.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada contém três inteiros N, P e E que representam o número de cidades, o número de pedreiras e o número de possíveis trechos a serem pavimentados respectivamente ($2 \leq N \leq 250, 1 \leq P \leq 20, 1 \leq E \leq \frac{1}{2}N^2$). As cidades são numeradas de 1 a N. A linha seguinte contém P inteiros em ordem crescente, indicando o comprimento das pedras produzidas pelas P pedreiras contratadas. Nenhuma pedra tem comprimento maior que 100. As E linhas seguintes contém três inteiros cada, U, V e T indicando que um trecho de comprimento T ($1 \leq T \leq 100$) ligando a cidade U à cidade V pode ser pavimentado. O tempo total nunca é maior que 2 bilhões.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo o menor tempo possível para se pavimentar os trechos de estrada necessários ou -1 caso não seja possível pavimentá-las de acordo com as restrições acima.

Entrada	**Entrada**	**Entrada**
3 4 2 1 2 3 4 1 2 10 2 3 5 **Saída** 29	4 3 6 2 5 10 1 2 1 1 3 3 1 4 20 2 3 10 2 4 3 3 4 2 **Saída** 10	5 2 7 13 19 1 2 12 2 3 15 3 4 8 3 5 14 1 4 3 1 5 19 4 5 13 **Saída** -1