In a country, there are N cities and M bi-directional roads. Some of the roads of the country are broken and need repairing. The king of the country wants a good transportation system, so he wants that all the cities of the country must be connected i.e. there must be at least one path to reach a city from any other city.

The king is also low on budget, so he wants to repair the roads in such a way that the cities of the country must be connected and the cost of repairing is as minimal as possible.

You have to find the minimum cost of repairing the roads such that the cities become connected.

Input Format:

First line of each test case contains two integers N and M denoting the no. of cities and no. of bi-directional roads. Each of next M lines contains the description of a road in following format:

U V 0

OR

U V 1 X

First two integers U and V denote the cities that are getting directly connected by this road. If third integer is 0, it means the road is OK and needs no repairing. If the third integer is 1, it means the road needs repairing and the cost of repairing that road is denoted by a fourth integer X.

Output Format:

For each test case, output a single integer denoting the minimum cost of repairing in order to make the cities connected.

Note:

1. The input always guarantees that there is at least a way to make all the cities of country connected.

2. There is at most one road between any two distinct cities.

3. There is no road from a city to itself.

Constraints:

1 <= N <= 10000

1 <= M <= 100000

1 <= U,V <= N

1 <= X <= 1000

Examples:

Input:

4 6

1 2 0

1 3 1 4

1 4 1 1

2 3 1 2

2 4 1 5

3 4 1 3

Output:

3

You have a grid of size N x M. Rows of the grid are numbered from 1 to N from top to bottom and columns of the grid are numbered from 1 to M from left to right. So, top-left corner is indexed as (1,1) and bottom-right corner is indexed as (N,M).

Cost of visiting a cell at index (i,j) is denoted by C[i][j]. Cost of changing the direction of your facing at index (i,j) is denoted by P[i][j].

You have to start from cell (1,1) and reach to cell (N,M) with minimum cost. At cell (1,1), you can either face towards 'Right' or face towards 'Down'. At any point of time, you can either face 'Right' or face 'Down'.

There are 2-types of moves allowed:

1. Move one cell towards the facing direction i.e. if you are facing 'Right', you can move one cell 'Right' or if you are facing 'Down', you can move one cell 'Down'. The cost of visiting cell (i,j) will be C[i][j].

2. Change the direction of facing i.e. if you are facing 'Right', you can now face 'Down' or if you are facing 'Down', you can now face 'Right'. The cost of changing the facing direction at cell (i,j) will be P[i][j].

Find the minimum cost to reach (N,M) after starting from (1,1).

Constraints:

1 <= T <= 3

1 <= N,M <= 1000

1 <= C[i][j], P[i][j] <= 1000

Input Format:

First line of each testfile contains T, the number of test cases.

In each testcase, First line contains two inetgers N and M denoting the dimensions of the grid.

Next N lines contains M integers each denoting the cost matrix C.

Next N lines contains M integers each denoting the cost matrix P.

Output Format:

For each test case, output a single line containing the minimum cost of reaching (N,M) from (1,1).

Example:

Input:

2

2 2

1 2

3 4

5 6

7 8

2 2

1 2

3 4

5 6

1 8

Output:

13

9

You are given two number-strings A and B. A number-string is a string that contains only digits ['0'-'9'].

The task is to make both the strings equal. The only operation that you are allowed to do is to delete a character (i.e. digit) from any of the strings (A or B).

The cost of an operation of deleting a digit D is D units.

Now, you have to make both the number-strings equal in minimum cost.

Constraints:

1 <= T <= 10

1 <= length(A),length(B) <= 1000

Input Format:

First line of each testfile contains T, the number of test cases.

In each testcase, first line contains number-string A and second line contains number-string B.

Output Format:

For each test case, output only line containing the minimum cost to make both number-strings equal.

Example:

Input:

3

4 3

7135

135

9 9

816337897

816393804

9 9

816337597

816393504

Output:

7

36

34