Problem Set for 2014 Taiwan Young-Students Programming Contest

September 19, 2014

This set contains 7 problems (A–G). Sample input data and the output for the sample input are given in each problem. You may create these sample input data to test your program. However, test cases used by the Judges may not be the same as the sample input.

It is not necessary to include comments in your programs. Any algorithm can be used to solve the problems. However, a program whose running time exceeds the time limit specified in each problem will be considered to be incorrect.

The input of each problem may contain more than one test case. The description of the input format is based on records. A record usually contains a list of data. Unless stated otherwise, these data may span more than one line. You may assume that the input data are always correct. That is, error checking of the input data is not required.

The input of each problem is the `standard input`. The output must be the `standard output`. For each problem, there should be an end-of-line at the end of the last line of the output.

Problem A
Cycle length in decimal expansion
Time Limit: *2 Seconds*

Given integers $p \leq q$ in which $p \geq 0$ and $q > 0$, find the cycle length in the decimal expansion of p/q. For example, $1/7 = 0.142857142857\ldots$, the cycle length is six; and $1/3 = 0.333\ldots$, so the cycle length is one. Note that $1/2 = 0.5$ which can also be expressed as $0.500\ldots$, and we assume the length is one.

Technical Specification

1. For each test case, we assume that $0 \leq p \leq q \leq 100000$.

Input File Format
The input consists of several cases. Each case contains two integers p and q in one line. The case with $q = 0$ ends the input, and you don't need to compute this case.

Output Format
For each case, print the cycle length in one line.

Sample Input

```
1 7
2 3
1 1
2 0
```

Output for the Sample Input

```
6
1
1
```

Problem B
Super gun
Time Limit: *2 Seconds*

There are n targets in the plane and you have a super gun. Your position is at coordinate $(0, 0)$. If you aim the super gun at a target and make one shot, all targets one the same line will be destroyed, no matter with the distances. Given the coordinates of the targets, your goal is to compute how many shots is needed to destroy all the targets.

Technical Specification

1. The number of targets n is at most 500000 in each test case.
2. The coordinate of each target is (x_i, y_i) for $1 \leq i \leq n$, where x_i, y_i are integers and $1 \leq x_i, y_i \leq 10000$. Two targets may be at the same position.

Input File Format
The first line contains an integer T indicating the number of test cases. For each test case, the first line is the number n. Each of the next n lines contains the coordinate x_i and y_i of one target, and the two integers are separated by a space.

Output Format
For each case, print the number of shots in one line.

Sample Input

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

Output for the Sample Input

```
2
3
```

Problem C
Domination
Time Limit: *2 Seconds*

A domination set of a graph is a node subset such that every node is either in the domination set or adjacent to at least one node in the set. Finding domination set of minimum total weight in general graphs is an NP-hard problem. However, it can be efficiently computed by observing a recurrence relation if the input graph is a path. More conveniently, we can describe the problem by a 1-dimensional array.

Consider an array A of n positive integers. A set of elements is a dominating set if for each element not in the set, there is at least one of its neighbors in the set. The goal is to choose a dominating set of minimum sum.

Technical Specification

1. The number of elements n satisfies $1 \leq n \leq 100000$ in each test case.
2. Each element is a positive integer at most 1000.

Input File Format
The first line contains an integer T indicating the number of test cases. For each test case, the first line is the number n, and the next line contains n integers which are the elements in the order. Two integers are separated by a space.

Output Format
For each case, print the minimized sum of any dominating set in one line.

Sample Input

```
2
5
10 20 30 40 50
3
1 5 3
```

Output for the Sample Input

```
50
4
```

Problem D
Transportation Network
Time Limit: *3 Seconds*

There are n cities and we want to build some airports and highways to connect these cities. For any two airports, there will be a direct flight. It is required that people can travel between any pair of cities via these highways and flights. All the highways will be bidirectional. Due to some reasons, the number of airports should be exactly k, and we don't care the cost of building the airports. Your job is to compute the minimum total cost of building the highways.

As a simple example, if there are four cities and exactly two airports ($n = 4$ and $k = 2$), then at least two highways are needed. You are given the costs of highways. For some pairs of cities, there may be more than one possible cost, and for some pairs it is impossible to build a highway between them.

Technical Specification

1. Each integer between 0 and $n - 1$ represents one city, where $2 \leq n \leq 100000$ for each test case.
2. The cost of any highway is an positive integer at most 50000, and the solution (minimum cost) is at most 2^{31}.
3. The number of possible highways m is a positive integer at most 500000.
4. $2 \leq k < n$.

Input File Format
The first line of the input file contains an integer T which indicates the number of test cases. Each test case starts with three integers n, m, and k in one line. Followed this line, there are m lines, and each line contains three integers u_i, v_i, and c_i, which indicates that we can build a highway between cities u_i and v_i with cost c_i. Two consecutive numbers in the same line are separated by a space.

Output Format
For each test case, output the minimum total cost of building highways in one line. If it is impossible, then output -1.

Sample Input

```
2
7 6 2
0 1 10
1 2 20
2 3 30
```

```
3 4 40
4 0 20
5 6 20
6 4 2
0 1 1
0 2 5
1 2 6
4 5 5
```

Output for the Sample Input

```
100
-1
```

Problem E
Maximum Difference Within a Range
Time Limit: *2 Seconds*

You are given an integer sequence $A = a_1, a_2, \ldots, a_n$ and a range bound L. Your job is to find the maximum difference for numbers in A such that the difference of indices of these two numbers is within the bound L. That is, find $\max_{i,j} |a_i - a_j|$ for $1 \leq i, j \leq n$ and $|i - j| \leq L$.

Technical Specification

1. $n \leq 5 \cdot 10^5$ and $1 \leq L < 5 \cdot 10^5$.
2. $-10^6 < a_i < 10^6$.

Input File Format
The input will consist of several input cases. The first line of each test case contains two numbers n and L, as described above, and they are separated by a space. The next line consists of n space-separated integers a_i's that represent the sequence A. An input case with $n = L = 0$ indicates the end of the input.

Output Format
You have to print the required maximum difference for each case in the input. Format the output as shown below.

Sample Input

```
6 2
1 8 -1 10 7 4
5 3
-4 1 5 2 6
5 5
1 2 3 4 5
10 11
1 1 1 1 1 1 1 1 1 1
20 4
1 2 3 4 5 6 7 8 9 0 9 8 7 6 5 4 3 2 1 0
0 0
```

Output for the Sample Input

```
11
9
4
0
9
```

Problem F
Tree Size
Time Limit: *2 Seconds*

Trees are used to represent the evolutionary relationship of species. An evolutionary tree is a edge-weighted tree. Each leaf represents one species, and the distances on the tree represents the dissimilarities among species. An important issue in computational biology is to construct the evolutionary tree from the observed dissimilarities.

Let $N = \{1..n\}$. An $n \times n$ matrix M is a metric over N if it is symmetric, nonnegative, and $M[i, j] + M[j, k] \leq M[i, k]$ for any $i, j, k \in N$ (triangle inequality). A metric is a *tree metric* if it can be realized by a tree, i.e., there exists an edge-weighted tree T such that

1. the leaf set is N;
2. the weight of each edge is nonnegative;
3. and $d_T(i, j) = M[i, j]$ for any $i, j \in N$, where $d_T(i, j)$ is the shortest path length between i and j on the tree T.

For example, the following matrix is a tree metric. The corresponding tree is given in the figure.

$$M = \begin{pmatrix} 0 & 5 & 9 & 12 & 8 \\ 5 & 0 & 8 & 11 & 7 \\ 9 & 8 & 0 & 5 & 1 \\ 12 & 11 & 5 & 0 & 4 \\ 8 & 7 & 1 & 4 & 0 \end{pmatrix}$$

The size of a tree is defined to be the total weight of the tree edges. For a tree metric, it has been shown that the tree size is unique, i.e., it is impossible to find two trees of different sizes realizing the same tree metric. Design a program to compute the tree sizes of the given tree metrics. The following simple example may be helpful.

Fot the case of only two species, the tree has only one edge and the tree size is just the disance between the two species. Let's consider the case of three species a, b and c. Let T be the corresponding tree. Since T has three leaves, there is an internal node x. By definition, the path length $d_T(a, b) = M[a, b]$. Since x is a vertex on the path between a and b, all we need to do is to

determine the weight (length) of edge (x, c). Let $w(x, c)$ denote the weight of edge (x, c). We have

$$w(x, c) + w(x, a) = M[a, c]$$
$$w(x, c) + w(x, b) = M[b, c]$$
$$w(x, a) + w(x, b) = M[a, b]$$

Therefore, $w(x, c) = (M[a, c] + M[b, c] - M[a, b])/2$.

Technical Specification

1. $n \leq 100$.
2. Each element in the distance matrix is nonnegative and at most 10000.

Input File Format

The input consists of several test cases. The first line of each test case is a positive integer n. The following $n - 1$ lines represent the upper triangle of the tree metric, but the diagonal is not included. Each line is for one row, and elements are separated by spaces. The case with $n = 0$ indicates the end of the input and you don't need to deal with it. You may assume that the test data are all tree metrics, and it is not necessary to check them. Furthermore the tree sizes are all integers in the test cases.

Output Format

For each test case, output the tree size in an individual line.

Sample Input

```
5
5 9 12 8
8 11 7
5 1
4
4
15 36 60
31 55
36
0
```

Output for the Sample Input

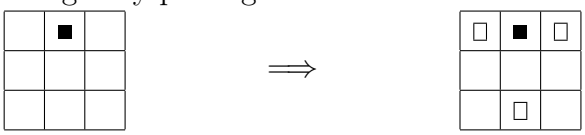
```
15
71
```

Problem G
A Generalized N-Queens Problem
Time Limit: *3 Seconds*

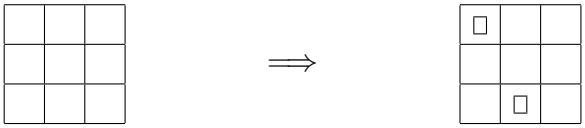
The following game is played on a masked N by N grid. A masked grid is one that some, if any, of the cells are pre-occupied with black stones. The goal of this single player game is to place as many white stones as possible to cells of a masked grid so that the following conditions hold at the same time:

1. You can place at most one white stone on a cell.
2. Two white stones cannot be placed so that they are in the same column, row or diagonal unless there is a black stone between them vertically, horizontally or diagonally, respectively.

Example 1: A 3 by 3 masked grid with one black stone on the left and a solution on the right by placing 3 white stones.



Example 2: A 3 by 3 masked grid without black stone on the left and a solution on the right by placing 2 white stones.



Your task is to write a computer program to compute the maximum number of white stones that can be placed on a given grid with some, if any, black stones.

Technical Specification

- N is an integer and $0 < N \leq 10$.
- The number of black stones can be 0 and is at most N^2.

Input Format

The first line of the input contains an integer, denoting the number of test cases, which is at most 10, to follow. For each test case, the first line contains N. In the following N lines, the ith line contains N characters which are the N numbers, separated by blanks, in the ith row where a number 1 means a black stone and a number 0 means an empty cell.

Output Format

For each test case, output in one line the maximum number of white stones

that can be placed.

Sample Input

```
2
3
0 1 0
0 0 0
0 0 0
3
0 0 0
0 0 0
0 0 0
```

Output for the Sample Input

```
3
2
```