ECOO 2019

Programming Contest Questions

Provincial Competition (Round 3)

May 11, 2019

Problem 1: Chords

Francesco has created his own piano-like instrument with **N** keys that each produce a unique frequency. One of his favourite things to do with the instrument is to play chords: a pair of keys where the frequency of one key is a multiple of the other. For example, the pair of frequencies (2, 4) is a pair, but (4, 6) is not.

Francesco does not want to keep playing the same chords all the time, so he would like to know how many different chords can be played using his instrument. Two chords are different if one contains a key that the other does not. Can you help him figure out how many chords can be played?

Input Specifications

DATA11.txt (DATA12.txt for the second try) will contain 10 datasets. Each dataset begins with a line containing an integer **N** ($1 \le \mathbf{N} \le 1{,}000{,}000$), the number of keys. The next line contains **N** distinct integers $\mathbf{F_i}$ ($1 \le \mathbf{F_i} \le 1{,}000{,}000$), the frequency of each key.

For the first 4 cases, **N** ≤ 1,000.

Output Specifications

For each dataset, output the number of chords that can be played.

Sample Input (Two Datasets Shown)

```
4
2 4 6 8
3
3 4 12
```

Sample Output

```
4
2
```

ECOO 2019 Question Development Team

John Ketelaars .... ECOO-CS Communications
David Stermole .... ECOO-CS President
Reyno Tilikaynen .... University of Waterloo
Sean Purcell .... University of Waterloo
Andrew Seidel .... John Fraser Secondary School, Mississauga
Seth Koehler .... Google
Tongbo Sui .... Google

Problem 2: Array Game

Alice and Bob have recently received a wonderful gift: a sorted array of positive integers! They have decided to create a new game using the array.

Starting with Alice, they take turns picking a positive element of the array and decreasing it by an arbitrary, non-zero amount. If a player cannot make a move on their turn, they lose. To make the game harder, they have also added the restriction that at the end of each player's turn, the array must still be sorted in non-decreasing order.

Alice and Bob have been playing this game for hours, but they are not sure if the moves that they are making are optimal. Hence, they have asked you to analyze their games and see who should have won if both players played optimally.

Input Specifications

DATA21.txt (DATA22.txt for the second try) will contain 10 datasets. Each dataset begins with an integer **T** ($1 \le T \le 5$), the number of games they wish to analyze. **T** game descriptions follow. Each game description consists of two lines. The first line contains an integer **N** ($1 \le N \le 16$), the number of elements in the array. The next line contains **N** integers A_i ($1 \le A_i \le 10$), the original array. The array is guaranteed to be sorted in non-decreasing order.

For the first 3 datasets, **N**, $A_i \le 5$.
For the first 6 datasets, $N \le 5$.

Output Specifications

For each dataset, output a string of length **T**. The **i**-th character should be 'A' if Alice should have won that game or 'B' if Bob should have won that game.

Sample Input (Two Datasets Shown)

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

Sample Output

```
A
AB
```

Explanation of Sample Datasets

In the first dataset, Alice can win the game by decreasing the single array value to zero.
In the second dataset, the only valid move in each turn of the first game is to decrease the left-most '1' to zero. Since there are an odd number of terms, Alice will win. In the second game, Alice must decrease the first element. If Alice decreases it to zero, Bob can decrease the other element to a zero. If Alice decreases it to 1, Bob can decrease the second element to 1, which is a winning position for him.

Problem 3: Wall

Thousands of years ago, the kingdom of Westeros decided to build a wall to deter invaders. The wall was an incredible feat of engineering, but it proved to have a major weakness: there are long stretches of the wall with no villages nearby. For these stretches of the wall, it was difficult to supply the guards and hence difficult to respond to attacks.

In the present day, historians who have been studying the wall wonder if it could have been built in a way that mitigates this weakness. Specifically, they are interested in the minimum number **D** such that one could build a wall where the length of each wall section between two villages is at most **D**.

The land of Westeros can be mapped on a 2-D grid. There are **N** villages located at integer coordinates on the grid. The wall starts at a given start village and can run between adjacent coordinates until it reaches some given end village. Coordinates are adjacent if their X and Y values differ by at most one (so up/down/left/right and diagonally). The length of the wall between adjacent coordinates is one unit (even diagonally).

The wall does not have to visit every village, but passing through villages between the start and end may reduce the length **D**.

Input Specifications

DATA31.txt (DATA32.txt for the second try) will contain 10 datasets. Each dataset begins with an integer **N** ($2 \le N \le 100{,}000$), the number of villages. The next **N** lines each contain two integers X_i, Y_i ($0 \le X_i, Y_i \le 10^9$), the coordinates of each village. The wall should start at the first village and end at the last village.

For the first 3 datasets, **N** ≤ 50.
For the first 6 datasets, **N** ≤ 1000.

Output Specifications

For each dataset, output the value **D**, the minimum possible distance between two consecutive villages.

Sample Input (Two Datasets Shown)

```
2
0 0
5 5
4
0 3
2 7
2 0
4 3
```

Sample Output

```
5
3
```

Explanation of Sample Datasets

In the second dataset, the wall should pass through the third village, which is at a distance of three units from both the start and end villages. Going from the start to either the second or fourth village has a distance of four units.

Problem 4: Network

Naomi works in a data center and has designed a new computer broadcast network which has the following structure:

- There are **N** machines in total.
- One of the machines is denoted as the “master” and is responsible for broadcasting new data.
- There are **N**-1 one-directional wires between machines in the network arranged in a way such that the master machine can contact all the other machines.

Such a network would require minimal infrastructure and allow the master machine to broadcast information while being insulated from the other machines.

Unfortunately, the interns tasked with implementing the network made two mistakes. First, they mixed up the directions of some of the one-directional wires. Second, they used a low-quality wire with high signal loss, which means that a given machine can only communicate with machines at most **R** connections away.

Nevertheless, Naomi remained optimistic: if she assigns multiple machines to the master role in such a way that each machine is at most **R** connections away from a master machine, then the network can still function. As Naomi’s star intern, can you help her figure out the minimum number of master machines that are required?

Input Specifications

DATA41.txt (DATA42.txt for the second try) will contain 10 datasets. Each dataset begins with two integers **N**, **R** ($1 \leq$ **N, R** $\leq 100{,}000$), the number of machines and the maximum distance that each machine can be from a master machine. Machines are numbered from 1 to **N**. The next **N**-1 lines each contain two distinct integers $\mathbf{A_I}$**,** $\mathbf{B_I}$ ($1 \leq \mathbf{A_i}, \mathbf{B_i} \leq \mathbf{N}$) stating that there is a one-directional wire from $\mathbf{A_I}$ to $\mathbf{B_I}$.

For the first 3 cases, **N** ≤ 20.
For the first 6 cases, **N** ≤ 100.

Output Specifications

For each dataset, output the minimum number of master machines required to make the network operational.

Sample Input (Two Datasets Shown)

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

Sample Output

```
1
2
```