2022 HDU Multi-University Training Camp
Contest 10

August 18, 2022

Important Notice

There should be 2 pictures attached to this PDF file, check if you have received them as well. One is for Problem 2 and the other is for Problem 6.

1 Winner Prediction

1.1 Problem Description

A tournament consisting of n participants is currently taking place. The players are numbered from 1 to n. Every match is between two participants, and there are no draws. The participant who wins the most matches wins the entire tournament; if there are multiple participants tied at the first place, all of them win the tournament.

At the current state, some matches have ended, and others are yet to start. You are given the results of all ended matches. Write a program to determine whether it is possible for player 1 to win the tournament.

You are given T independent test cases. Solve each of them.

1.2 Input

The first line of input consists of a single integer $T(1 \le T \le 100)$, indicating the number of test cases. Then T test cases follow.

Each of the T test cases consists of multiple lines.

The first line contains three integers $n, m_1, m_2(1 \le n \le 500, 1 \le m_1, m_2 \le 1000)$, indicating the number of participants, the number of ended matches and the number of upcoming matches.

Each of the next m_1 lines contains three space-separated integers $x, y, z(1 \le x, y \le n, x \ne y, 0 \le z \le 1)$, indicating an ended match between player x and player y , $z = 1$ means player x won the match and $z = 0$ means player y won the match.

Each of the next m_2 lines contains two space-separated integers $x, y(1 \le x, y \le n, x \ne y)$, indicating an upcoming match between player x and player y.

1.3 Output

For each test case, if it is possible of player 1 to win the tournament, print a line `YES`; otherwise print a line `NO`.

1.4 Sample Input

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

1.5 Sample Output

```
YES
NO
```

2 Photos

2.1 Problem Description

You have a square farm, divided into $n \times n$ cells. There are m beautiful cells among all cells, whose locations are given. You would like to take some photos of the farm. A photo should figure a square area of the farm, with vertices located on grid points and edges parallel to the grid lines. What's more, its principal diagonal should lie on the principal diagonal of the whole farm.

You would like to take several photos so that each beautiful cell is contained in one photo. You hate repetitions so that any two of your photos should not coincide, that is, their common area is 0. Each photo can contain an arbitrary number of beautiful cells, possibly 0, 1 or more. Count the number of set of photos satisfying the above conditions, modulo 998244353.

In case you are confused, here we provide an illustration. The farm shown in this illustration is the same as the one in the sample test case.

You are given T independent test cases; solve each of them.

2.2 Input

The first line of input contains a single integer $T(1 \leq T \leq 1500)$, indicating the number of test cases. Then T test cases follow.

The first line of each test case contains two space-separated integers $n, m(1 \leq n \leq 10^9, 1 \leq m \leq 10^5)$, denoting the size of the farm and the number of good cells. Each of the next m lines contains two space-separated integers $x_i, y_i(1 \leq x_i, y_i \leq n)$, denoting the coordinates of a good cell.

It is guaranteed that $\sum m \leq 10^6$.

2.3 Output

For each test case, print a line consisting of a single integer, indicating the answer modulo 998244353.

2.4 Sample Input

```
1
6 2
4 2
3 4
```

2.5 Sample Output

```
24
```

3 Wavy Tree

3.1 Problem Description

An array a of length n is said to be wavy, if for each $1 < i < n$ either $a_i > \max\{a_{i-1}, a_{i+1}\}$ or $a_i < \min\{a_{i-1}, a_{i+1}\}$ holds.

You are given an array b of length n $(1 \leq b_i \leq 10^9)$, consisting of integers. You want to make the array wavy. To do that you can spend some coins, with each coin you can make one element in b increase or decrease by 1. Calculate the minimum number of coins you need to spend to make the array wavy.

3.2 Input

The first line contains the number of test cases T $(1 \leq T \leq 10^3)$.

The first line of each test case contains one integer n $(1 \leq n \leq 10^6)$ - the length of array b .

The second line contains n integers $b_1, b_2, \cdots, b_n$ $(1 \leq b_i \leq 10^9)$ - the array b .

It's guarantee that the sum of n among all test cases is not greater than 3×10^6 .

3.3 Output

For each test case, output one integer, the minimum number of coins you need to spend to make the array wavy.

3.4 Sample Input

```
3
4
1 7 6 5
6
1 2 3 4 5 6
6
1 1 4 5 1 4
```

3.5 Sample Output

```
2
4
4
```

4 Average Replacement

4.1 Problem Description

There are n people in a group and m pairs of friends among them. Currently, each person writes an integer on his hat. They plan to play the following game many times: everyone replaces his number on his hat with the average number of his number and all of his friends' numbers. That is, if before the game the person has a_0 written on his hat and a total of k friends, each having number $a_1, ..., a_k$, then after the game the number on his hat becomes $(a_0 + \cdots + a_k)/(k+1)$. Note that numbers may become non-integers.

It can be proved that by playing more and more games, each number converges to a certain value. Given the initial numbers written on the hats, your task is to calculate these values.

4.2 Input

The first line contains the number of test cases T $(1 \leq T \leq 100)$.

For each test case, the first line contains two integers n, m $(1 \leq n, m \leq 10^5)$.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(1 \leq a_i \leq 10^8)$, indicating the number on each hat.

Each of the following m lines contains two integers u, v $(1 \leq u, v \leq n)$, indicating a pair of friends.

It's guaranteed that there are no self-loop or multiple edges on the graph, and there are at most 20 test cases such that $n > 1000$ or $m > 1000$.

4.3 Output

For each test case, output n real numbers in n lines, indicating the value of each person at last. The results should be reserved with 6 digits after the decimal point.

4.4 Sample Input

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

4.5 Sample Output

```
1.500000
1.500000
1.500000
1.500000
3.500000
3.500000
```

5 Apples

5.1 Problem Description

The people in city A want to share their apples. There are n people in city A, and they are numbered from 1 to n. The i-th person has b_i apples initially, and this person will be happy if and only if he/she has exactly e_i apples after sharing.It is guaranteed that the total number of apples will not changed, which is $\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} e_i$.

The city has n undirected roads. The i-th road connects the i-th and the $(i \bmod n + 1)$-th person's house. the apples can be transported by these roads. Each road has a value l_i, denoting the cost of transporting a single apple from person i to $(i \bmod n + 1)$ or from person $(i \bmod n + 1)$ to i. Each apple can be transported by any road any number of times(including zero).

You need to find out a way to make all the people become happy and minimize the total cost of all the apples. And the cost of an apple is the total cost of all the roads it passed. Noted that an apple can pass the same road more than one time, and the cost will be counted repeatedly.

The cost of some roads may be changed, and you need to find out all the answers.

5.2 Input

The first line contains a single integer $T (T \leq 100)$ - the number of test cases.

For each test case:

The first line contains a single integer $n (2 \leq n \leq 5 \times 10^5)$ - the number of people.

From the second line to the $(n + 1)$-th line, each line contains three integers $b_i, e_i, l_i (0 \leq b_i, e_i \leq 10^9, 0 \leq l_i \leq 10^4)$. It is guaranteed that $\sum_{i=1}^{n} b_i = \sum_{i=1}^{n} e_i \leq 10^9$.

The $(n + 2)$-th line contains a single integer $q (0 \leq q \leq 5 \times 10^5)$.

Then the next q lines, each line contains two integers $x, y (1 \leq x \leq n, 0 \leq y \leq 10^4)$, means that l_x is changed to y. Please note that the change in l_x has aftereffects.

It is guaranteed that the sum of n and the sum of q do not exceed 2×10^6.

5.3 Output

For each test case, output $(q + 1)$ lines:

Each line contains a single integer, The first integer denotes the answer without any changing, and the $(i+1)$-th integer $(1 \leq i \leq n)$ denotes the answer after the i-th change.

5.4 Sample Input

```
2
4
```

```
4 1 4
5 1 4
1 9 1
9 8 1
0
2
1 2 1
2 1 2
3
1 3
1 2
2 1
```

5.5 Sample Output

```
23
1
2
2
1
```

6 Triangle Rotation

6.1 Problem Description

You are given a triangle tower of n layers. There are i vertices in the i-th layer, and at each vertex there is an integer written on it.

Below is a figure for $n = 4$.

It can be shown that there are a total of $n(n+1)/2$ vertices. We guarantee that the numbers are a permutation of all integers in $[1, n(n+1)/2]$.

You need to sort the numbers, first by row and second by column, with some numbers of **triangle rotations**. A triangle rotation means:

- Select a unit triangle (the smallest non-zero triangle you can find in the figure) and rotate the numbers on its three vertices **clockwise**.

Determine whether there exists a way to sort the numbers within $2n^3$ operations. If yes, print out one of them.

6.2 Input

The first line contains an integer $T(1 \leq T \leq 150)$ - the number of test cases.

The first line of each test case contains an integer $n(2 \leq n \leq 50)$ - the number of layers of the tower.

The next n lines of each test case represent the numbers in the tower. The i-th line contains i numbers.

It is guaranteed that $\sum n^3 \leq 10^6$.

6.3 Output

For each test case, Output ”Yes” or ”No” in a single line, indicating whether there exists a way to sort the numbers within $2n^3$ operations.

If your answer is ”Yes”, Output an integer $k(0 \leq k \leq 2n^3)$ - the number of operation you used in a single line.

For the next k lines, output three integers $x, y(1 \leq x \leq n-1, 1 \leq y \leq 2x-1)$, indicating an operation at the y-th triangle between the x-th layer and the $x+1$-th layer.

6.4 Sample Input

```
3
3
6
4 5
2 1 3
2
2
1 3
2
```

```
2
3 1
```

6.5 Sample Output

```
Yes
11
2 3
1 1
1 1
2 3
2 3
2 2
2 1
2 1
2 2
2 3
2 3
No
Yes
2
1 1
1 1
```

7 Even Tree Split

7.1 Problem Description

You are given an undirected tree with n nodes. It's guaranteed that n is even.

You are going to delete some of the edges (at least 1), and have to let each of the remaining connected components have an even number of vertices.

Calculate the number of ways to delete the edges that satisfy such constraints, modulo 998244353.

7.2 Input

The first line contains an integer $T(1 \leq T \leq 30)$ - the number of test cases.

The first line of each test case contains an integer $n(1 \leq n \leq 10^5)$ - the number of vertices on the tree.

The next $n-1$ lines of each test case contain two integers $u, v(1 \leq u, v \leq n)$, representing an edge between u and v.

It is guaranteed that the input graph is a tree with even number of vertices.

7.3 Output

For each test case, output the number of ways to delete the edges that satisfy such constraints in a single line, modulo 998244353.

7.4 Sample Input

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

7.5 Sample Output

```
0
1
```

8 Minimum Diameter

8.1 Problem Description

The following is the **minimum diameter problem**.

- You are given a forest (an acyclic undirected graph) with n vertices. Consider adding some edges to the forest to turn it into a tree. Find the minimum possible diameter of the resulting tree.

Here the diameter of a tree is defined as the maximum distance among all pairs of vertices. The distance of two vertices in a tree is defined as the number of edges on the shortest path between them.

You are given a forest of n vertices and m edges. The edges are numbered from $1, 2, ..., m$. For each $i = 1, 2, ..., m$, consider the forest only containing the first i edges, and compute the answer to the **minimum diameter problem** on this forest.

8.2 Input

The first line contains a single integer T $(1 \leq T \leq 10^3)$ - the number of test cases.

For each test case, the first line contains two integers n, m $(2 \leq n \leq 10^5, 1 \leq m < n)$.

Each of next m lines contains two integers u and w $(1 \leq u, w \leq n)$ - describes the i-th edge of the forest.

It's guarantee that the sum of n among all test cases is not greater than 10^6 and m edges form a forest.

8.3 Output

For each test case, output m lines. The i-th of these lines should contain a single integer, indicating the answer to the **minimum diameter problem** on the forest only containing the first i edges of the original forest.

8.4 Sample Input

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

8.5 Sample Output

```
2
2
```

```
3
4
```

9 Painting Game

9.1 Problem Description

There is a paper strip divided into n blank grids. For each $i (1 \leq i < n)$, grid i and $i + 1$ are considered adjacent.

Alice and Bob are going to play a game on the strip. They take turns to make move. In one move the player must paint one of the remaining blank grids black, while keeping the rule that no two black grids are adjacent.

The game ends when one of the players is unable to paint any grid, and the score of the game is defined as the total number of grids painted black. Alice wants to minimize the score, while Bob wants to maximize it.

Given n and the side starting the game, find out the final score when both players play optimally.

9.2 Input

The first line contains an integer $T (1 \leq T \leq 10^5)$ - the number of test cases.

The first line of each test case contains an integer $n (1 \leq n \leq 10^9)$ and a string $s (s \in \{\texttt{Alice}, \texttt{Bob}\})$ - the number of grids and the starting player of this game.

9.3 Output

For each test case, output the final score when both players play optimally in a single line.

9.4 Sample Input

```
4
3 Alice
3 Bob
19 Alice
23 Bob
```

9.5 Sample Output

```
1
2
8
10
```

10 Tree

10.1 Problem Description

You are given a directed graph with n vertices and m edges. The vertices are numbered from 1 to n.

For each vertex i, find out the number of ways to choose exactly $n-1$ edges to form a tree, where all the other $n-1$ vertices can be reached from i through these $n-1$ edges.

10.2 Input

The first line contains a single integer $T(1 \le T \le 100)$ - the number of test cases.

For each test case:

The first line contains two integers $n, m(1 \le n \le 500, 0 \le m \le n \times (n-1))$ - the number of vertices and the number of edges.

The next m lines, each line contains two integers $x, y(1 \le x, y \le n, x \ne y)$, denoting an edge. It is guaranteed that all the edges are different.

It is guaranteed that there are no more than 3 test cases with $n > 100$.

It is guaranteed that there are no more than 12 test cases with $n > 50$.

10.3 Output

For each test case, output n integers in a line, the i-th integer denotes the answer for vertex i. Since the answer may be too large, print it after modulo 10^9+7.

Please do not have any space at the end of the line.

10.4 Sample Input

```
2
1 0
7 12
1 3
2 1
1 4
5 1
4 7
6 5
2 3
4 6
3 1
6 4
7 1
1 2
```

10.5 Sample Output

```
1
2 3 1 4 2 6 2
```

11 Maximum Triangles

11.1 Problem Description

We called a triangle is good if and only If the triangle contains the origin.

You need to find n points on the plane, satisfying:

- None any two of them and the origin should be collinear.

- The coordinates of each point should be an integer and in the range $[-50000, 50000]$.

- Under the above limits, the number of good triangles made up of those n points should be maximized.

Output the maximum number and a set of the coordinates of those n points for which the maximum is reached.

11.2 Input

The first line of input contains a single integer $T(1 \le T \le 10)$, indicating the number of test cases.

Each of the next T lines contains a single integer $n(1 \le n \le 2 \times 10^5)$, describing the number of points you have to find for that test case.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

11.3 Output

For each test case print $(n+1)$ lines. The first line should contain a single integer, denoting the maximum number of good triangles. The i-th of the next n lines should contain two space-separated integers $x_i, y_i(|x_i|, |y_i| \le 50000)$, denoting the coordinates of the i-th point of the set. If there are multiple solutions, output any.

11.4 Sample Input

```
1
3
```

11.5 Sample Output

```
1
0 1
-1 -1
1 -1
```

12 Expected Inversions

12.1 Problem Description

For an integer sequence $a_1, ..., a_n$ of length n, its inversion number $\text{inv}(a)$ is defined as the number of integer pairs (i, j) such that $1 \leq i < j \leq n$ and $a_i > a_j$.

For a given rooted tree of n nodes (with vertices numbered from 1 to n), a **DFS procedure** on the tree is as following.

- During the process, we maintain a **current vertex**, namely u, and a **set of visited vertices**, namely M.

- Let the root of the tree be x. Initially, $u = x$ and $M = \{x\}$.

- Repeat the following process until M contains all vertices:

- - If there is at least one child vertex of u that is not in M, randomly choose one among those vertices equiprobably (namely v). Set u to v and add v to M.

- Otherwise, set u to the father of u.

For each $u = 1, ..., n$, we record the number of vertices in M when u is added to M (including u). Let this number be d_u. We call $(d_1, d_2, ..., d_n)$ a **DFS order**. As **DFS procedure** is non-deterministic, the resulting **DFS order** may vary as well. Assume that each decision in the **DFS procedure** is independent.

You are given an unrooted tree of n vertices, with vertices numbered from 1 to n. For each $i = 1, ..., n$, compute the expected inversion number of the **DFS order** when rooting the tree at i and start a **DFS procedure**. To avoid precision errors, print the answer modulo 998244353.

You are given T independent test cases. Solve each of them.

How to compute non-integers modulo 998244353: It can be proved that the answer to this problem can always be written as a fraction P/Q with P, Q being integers and $Q \not\equiv 0 \pmod{998244353}$. There is exactly one integer $R \in [0, 998244353)$ that satisfies $QR \equiv P \pmod{998244353}$. Print this R as the answer.

12.2 Input

The first line of input contains a single integer $T(1 \leq T \leq 10)$, indicating the number of test cases. Then T test cases follow.

The first line of each test case contains a single integer $n(1 \leq n \leq 10^5)$, indicating the number of vertices in the tree. Each of the next $n - 1$ lines contains two integers $u, v(1 \leq u, v \leq n)$, indicating an edge on the tree. It is guaranteed that the input edges form a tree.

12.3 Output

For each test case, print the answers in n lines. The i-th line should contain the expected inversion number of the **DFS order** when rooting the tree at vertex i.

12.4 Sample Input

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

12.5 Sample Output

```
499122177
1
2
```