2022 NTU ICPC Team Preliminary

National Taiwan University

2022/08/13

Language	Version	Compile Flags	Extensions
C++	g++ 9.4.0	`-O2 -std=c++17 -static`	`.cc, .cpp`

Problem	Problem Name	Time Limit	Memory Limit
A	Abstract	1 s	256 MB
B	Bipartite Graph	2 s	256 MB
C	Cactus? Cactus!	3 s	512 MB
D	Depth First Search	2 s	256 MB
E	Eccentric Game	1 s	256 MB
F	Finding the Area	1 s	256 MB
G	Game Night	1 s	256 MB
H	Hula Hoop	1 s	256 MB
I	Incredible Subsequence	2 s	512 MB
J	Just Operations	3 s	256 MB
K	King Game	3 s	256 MB
L	Loli	1 s	256 MB

This page is intentionally left blank.

A. Abstract

Problem ID: abstract

Hi, so to me seems like a notorious coincidence.

Given a directed acyclic graph with **exactly one terminal** (a vertex with no exiting edges) and numbers written on each vertex, please output the final value of the variable `result`, after `Count_Seconds()` is executed once:

```
function gao()
    for x in vertices
        b[x] = 0
        for y in vertices
            if an edge from y to x exists in the graph
                b[x] += a[y]
            Endif
        Endfor
        if x is the terminal
            b[x] += floor(a[x] / 2)
        Endif
    Endfor
    for x in vertices
        a[x] = b[x]
    Endfor
Endfunction

function Count_Seconds()
    result = 0
    while there exists x in vertices such that a[x] != 0
        result += 1
        gao()
    Endwhile
Endfunction
```

Input

The first line of the input contains two integers N, M indicating the number of vertices and the number of edges.

The second line contains N integers $a_1, a_2, \cdots, a_n$, denoting the integers on vertices.

M lines follow, the i-th of which contains two integers u_i, v_i, representing a directed edge from u_i to v_i.

- $1 \le N \le 10^4$
- $1 \le M \le 10^5$
- $1 \le u_i \le N$
- $1 \le v_i \le N$
- $0 \le a_i \le 10^9$
- The given graph is a DAG without multi-edges, and there is exactly one node that has no outgoing edges.

Output

Output an integer in one line – the value of `result` after one execution of `Count_Seconds()`. Or in other words, the first moment of time when all a_i become 0, modulo 998244353.

Sample Input 1

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

Sample Output 1

```
3
```

Sample Input 2

```
6 8
1 1 4 5 1 4
1 4
1 5
2 3
2 5
3 4
4 5
4 6
5 6
```

Sample Output 2

```
8
```

B. Bipartite Graph

Problem ID: bipartite

Given a bipartite graph with N and M vertices on each side (left and right) and K edges connecting them. The following method embeds the graph on a 2D plane:

- Place vertex $u_1, u_2, \ldots u_N$, vertices on the left side, at any N **distinct** points on the line $x = 0$.
- Place vertex $v_1, v_2, \ldots v_M$, vertices on the right side, at any M **distinct** points on the line $x = 1$.
- Each edge (u_i, v_i) represents a straight segment connecting u_i and v_i on the plane.

What is the minimum number of segment **non-endpoint intersections** of the embedding if the vertices are placed optimally?

Input

The first line of the input contains three integers N, M, K, indicating the number of vertices on the left side, the number of vertices on the right side, and the number of edges.

K lines follow, and the i-th of which contains two integers u_i, v_i, indicating that the i-th edge connects vertex u_i on the left side and vertex v_i on the right side.

- $1 \leq N + M \leq 16$
- $1 \leq K \leq N \cdot M$
- $1 \leq u_i \leq N$
- $1 \leq v_i \leq M$
- The given bipartite graph is simple

Output

Output an integer in a single line – the minimum number of segment intersections if the vertices are placed optimally.

Sample Input 1	**Sample Output 1**
3 2 5 1 1 1 2 2 1 2 2 3 1	1

Sample Input 2	**Sample Output 2**
3 4 5 1 4 3 4 3 2 3 1 2 3	0

C. Cactus? Cactus!

Problem ID: cactus

> In graph theory, a cactus (sometimes called a cactus tree) is a connected graph in which any two simple cycles have at most one vertex in common. Equivalently, it is a connected graph in which every edge belongs to at most one simple cycle.
>
> (Wikipedia)

Now, given an undirected cactus with N vertices and M edges. Each edge has a weight on it. Define the weight of a **simple path** as the **xor-sum** of the weights of the edges on the path.

You need to answer Q queries: what is the minimum possible weight of a **simple path** from u to v? Note that you cannot traverse a vertex twice in a simple path.

Input

The first line of the input contains three integers N, M, Q, indicating the number of vertices, the number of edges, and the number of queries.

M lines follow, and the i-th of which contains three integers u_i, v_i, w_i, indicating that the i-th edge connects vertex u_i and vertex v_i, having a weight w_i on it.

Last, followed by Q lines, each line contains two integers a_i, b_i, indicating the i-th query.

- $1 \leq N, Q \leq 10^5$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq u_i, v_i \leq N$
- $0 \leq w_i < 2^{30}$
- $1 \leq a_i, b_i \leq N, a_i \neq b_i$
- The given graph is simple
- The given graph is a cactus

Output

For each query, output the minimum weight of a simple path from u to v on a separate line.

Sample Input 1

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

Sample Output 1

```
0
2
2
```

Sample Input 2

```
9 11 7
1 2 3
1 3 2
2 3 5
2 4 1
3 5 4
5 6 6
3 6 7
6 7 8
7 8 1
7 9 3
8 9 2
1 2
1 5
2 6
3 4
4 9
5 7
5 8
```

Sample Output 2

```
3
2
2
0
8
11
10
```

D. Depth First Search

Problem ID: dfs

YP is learning graph theory. After learning about Depth-First Search, he wrote the following pseudo-code:

```
1 function dfs(u)
2     mark u as visited
3     output u
4     for v in vertices adjacent to u
5         if v is not visited, then
6             dfs(v)
7         Endif
8     Endfor
9 Endfunction
10
11 function run_dfs()
12     for v in all vertices
13         if v is not visited, then
14             dfs(v)
15         Endif
16     Endfor
17 Endfunction
```

YP noticed that there are some non-deterministic behaviors in line 4 and line 12. Specifically, the order of the traversed vertices might not be fixed!

Therefore, YP asks you the following problem: Given an undirected graph with N vertices and M edges and a permutation $p_1, p_2, \ldots, p_N$, what is the minimum number of edges you need to add to **make it possible** for the output order to become the permutation p? Moreover, you need to provide the added edges for YP to check your answer.

It can be shown that there always exists an answer to YP's problem.

Input

The first line of the input contains two integers N, M, indicating the number of vertices and the number of edges.

M lines follow, and the i-th of which contains two integers u_i, v_i, indicating that the i-th edge connects vertex u_i and vertex v_i.

The last lines contains N integer $p_1, p_2, \ldots, p_N$, indicating the given permutation.

- $1 \leq N \leq 3 \times 10^5$
- $0 \leq M \leq 5 \times 10^5$
- $1 \leq u_i, v_i \leq N, u_i \neq v_i$
- $p_1, p_2, \ldots, p_N$ is a permutation of $[1, N]$

Output

Output an integer K, the minimum number of edges you need to add, in the first line.

K lines follow, and the i-th of which contains two integers a_i, b_i, indicating the i-th edge you want to add which connects a_i, b_i, with $1 \leq a_i, b_i \leq N, a_i \neq b_i$.

If there are multiple answers, you can output any.

Sample Input 1

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

Sample Output 1

```
2
1 2
3 5
```

Sample Input 2

```
8 8
2 8
3 8
5 6
1 6
6 3
8 7
2 3
4 3
1 8 7 5 4 2 3 6
```

Sample Output 2

```
4
1 8
2 4
4 5
5 8
```

E. Eccentric Game

Problem ID: eccentric

Alice and Bob are playing the famous Nim game. Currently, they have a pile of stones. Each player take turns to remove at least one stone from the pile. Alice goes first. The player who can't make a move loses.

However, to make the game more interesting, and to make you more confused, they added two new rules as follows:

1. If there are n stones at the start of the game, Alice cannot remove more than m ($m \leq n$) stones in the first move.
2. The number of stones each player removes in a move cannot exceed twice the number of the last move from the other player. That is, if the other player removes k stones in the last move, then the current player cannot remove more than $2k$ stones in this move.

As you can see, a game is defined by a pair of positive integers (n, m). Given a positive integer N, find the number of pairs (n, m) such that $1 \leq m \leq n \leq N$ and Alice wins in the game defined by (n, m), if both players play optimally.

Input

The first line contains a single integer Q, the number of test cases in one input file. Each test case only contains a single line with a single integer N in it.

- $1 \leq Q \leq 10^4$
- $1 \leq N \leq 10^{18}$

Output

For each test case, output the answer modulo 998244353.

Sample Input 1	Sample Output 1
10	1
1	2
2	3
3	7
4	8
5	2184
69	86234
420	566188078
114514	481254019
998244353	411505598
1000000000000000000	

F. Finding the Area

Problem ID: finding

You are given N 2D vectors $v_1, v_2, \ldots, v_N$, consider the following set of 2D points S:

$$S = \{p \mid p = a_1 \cdot v_1 + \cdots + a_n \cdot v_n, 0 < a_1, ..., a_n < 1, a_i \in \mathbb{R}\}$$

Calculate the area of S.

Input

The first line of the input contain an integer N, indicating the number of vectors.

N lines follow, and the i-th of which contains two integers x_i, y_i, indicating that $v_i = (x_i, y_i)$.

- $1 \leq N \leq 3 \times 10^5$
- $-3000 \leq x_i, y_i \leq 3000$

Output

Output a real number in a single line – the area of S. Your answer will be considered correct if the absolute or relative error of it to the judges' answer does not exceed 10^{-9}.

Sample Input 1	**Sample Output 1**
2 0 1 1 0	1.000000000

Sample Input 2	Sample Output 2
10 -10 10 -2 -6 5 6 8 -3 -10 9 6 8 -9 5 6 -2 4 1 -2 6	2123.000000000

G. Game Night

Problem ID: gamenight

On Christmas Eve, you are going to play a game with Teacher Lin, Teacher Zhang, Teacher Tseng, Teacher Huang, Teacher Cheng, Teacher Chien and Teacher Tsai to determine how to distribute the remaining bottles of warm Coke.

In the game, each player should think of an ordered pair of integers that are both between 1 and M (inclusive) independently. On the count of 18000, all players should simultaneously post their pairs in a messenger group. For a player who posted (x, y) in the group, if there exists another player who posted (v, w) such that $v = x$ **or** $w = y$, both players lose!

If none of the players lose, they would have to think of another game such as Dinic's Algorithm Typing Contest to determine the losers. But in such case, you know for sure that you will be the one drinking all of the remaining warm Coke. Thus, you hope that all players lose the game, so that everyone only needs to drink a little warm Coke.

Before the game actually started, some teachers left and some other teachers joined. In the end, there are N players.

Please count the number of ways the N players can choose their pair, such that in the resulting game, all players lose. Two ways are considered different if at least one player chose a different pair between the two ways.

Input

The input contain two integer N, M, indicating the number of players and the largest number they can choose for their pairs.

- $2 \le N \le 10^6$
- $1 \le M \le 10^6$

Output

Output the number of different ways, modulo 998244353.

Sample Input 1	Sample Output 1
2 2	12

Sample Input 2	Sample Output 2
3 3	369

H. Hula Hoop

Problem ID: hula

YP is playing with a hula hoop. A hula hoop can be regarded as a circle with radius r. YP is standing at the origin of the 2D plane $(0,0)$. N obstacles are located at $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$. To help YP decide his hula hoop's radius, find the minimum radius of the hula hoop, L, such that the hula hoop has a chance to collide with at least one obstacle.

Suppose that the hula hoop is a circle with radius r, and its center is currently at (p, q). Since YP stands at $(0,0)$,

- $\sqrt{p^2 + q^2} = r$

The hula hoop collides with an obstacle (x_i, y_i) if:

- $\sqrt{(p - x_i)^2 + (q - y_i)^2} = r$.

Input

The first line of the input contains one integer N, indicating the number of obstacles.

N lines follow, and the i-th of which contains two integers x_i, y_i, indicating that the i-th obstacle is located at (x_i, y_i).

- $1 \le N \le 100$
- $-10^9 \le x_i, y_i \le 10^9$
- $(x_i, y_i) \ne (0,0)$

Output

Output a real number in a single line – the minimum radius L.

Your answer will be considered correct if the absolute or relative error of it to the judges' answer does not exceed 10^{-9}.

Sample Input 1	Sample Output 1
3 1 1 3 -2 -2 0	0.707106781186548

I. Incredible Subsequence

Problem ID: incredible

Increasing subsequence? No, let's have more fun with another subsequence problem.

An interval $[l, r]$ of a sequence $a_1, a_2, \ldots, a_n$ is called a *flank interval* if the unordered set $\{a_l, a_r\} = \{m_{l,r}, M_{l,r}\}$, where $m_{l,r}$ and $M_{l,r}$ are the minimum and the maximum of $a_l, a_{l+1}, \ldots, a_r$, respectively. A sequence $a_1, a_2, \ldots, a_n$ is called an *incredible sequence* if there are no flank intervals of it with length at least 3. For example, $1, 4, 2, 3$ is an incredible sequence, while $3, 1, 2, 4$ is not.

You are given a permutation $p_1, p_2, \ldots, p_N$ of $1, 2, \ldots, N$. Please count the number of non-empty subsequences that are incredible sequences. Since the answer may be very large, please output the answer modulo $10^9 + 7$.

Input

The input consists of two lines. The first line contains only one integer N, denoting the length of the permutation.

The second line contains N integers $p_1, p_2, \ldots, p_N$. It is guaranteed that the input is a permutation of $1, 2, \ldots, N$.

- $1 \leq N \leq 3000$

Output

Please output an integer denoting the answer. Since the answer may be very large, please output it modulo $10^9 + 7$.

Sample Input 1	Sample Output 1
3 1 2 3	6

Sample Input 2

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

Sample Output 2

```
29
```

Sample Input 3

```
10
6 8 4 10 9 3 5 2 1 7
```

Sample Output 3

```
245
```

J. Just Operations

Problem ID: just

In this problem, you just need to perform Q operations on two integer sequences $a_1, a_2, \ldots, a_N$ and $b_1, b_2, \ldots, b_N$. Each of the operations may be one of the three following kinds:

- $1\ l\ r\ v$: for all $l \le i \le r$, $a_i := a_i + v$
- $2\ l\ r\ v$: for all $l \le i \le r$, $b_i := b_i + v$
- $3\ l\ r$: find the maximum of $a_i \cdot i + b_i$ for all $l \le i \le r$

Input

The first line of the input contains two integers N, Q, indicating the length of the sequences and the number of operations.

The second line of the inputs contains N integers $a_1, a_2, \ldots, a_N$, indicating the first sequence.

The third line of the inputs contains N integers $b_1, b_2, \ldots, b_N$, indicating the second sequence.

Q lines follow, and the i-th of which contains one of the given operations described in the problem description, indicating the i-th operation.

- $1 \le N, Q \le 2 \times 10^5$
- $1 \le a_i, b_i, v \le 10^6$
- $1 \le l \le r \le N$

Output

For each type 3 operation, output the answer on a separate line.

Sample Input 1

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

Sample Output 1

```
15
19
20
```

Sample Input 2

```
10 10
2 3 3 4 2 4 1 4 5 4
4 5 2 1 3 4 1 1 2 1
1 7 8 2
2 2 5 1
1 8 9 1
3 2 3
2 6 9 2
3 7 8
1 6 8 1
3 6 9
1 4 7 10
3 1 10
```

Sample Output 2

```
12
59
67
101
```

K. King Game

Problem ID: kinggame

YP and PY are playing King Game again. This time, they decide to play another version of king game.

King game is a fantastic poker game that originated from National Taiwan University. It's played with a standard deck of 52 cards with four suits (spades, hearts, diamonds, and clubs) and 13 ranks(A, K, Q, J, and 10-2) without jokers.

The game goes as described below.

In the beginning, YP and PY are dealt with four cards face down. These cards are called **hole cards** and should not be revealed until the showdown. Then the dealer deals five cards face up as community cards.

After those five cards are dealt, the game goes to the showdown. All players reveal their hole cards at this point. Then each player chooses **two** out of their own four hole cards and **three** out of the five community cards to create a **hand**. The player creating the strongest hand wins the game. There are ten possible kinds of hands, listed from the strongest to the weakest as follow:

- Royal straight flush: A, K, Q, J and 10 of the same suit. This is a special case of straight flush.
- Straight flush: Five cards in sequence (e.g. 7, 6, 5, 4 and 3) and of the same suit.
- Four of a kind: Four cards of the same rank.
- Full house: Three cards of the same rank, plus a pair of another rank.
- Flush: Five cards of the same suit, but not in sequence.
- Straight: Five cards in sequence, but not of the same suit.
- Three of a kind: Just three cards of the same rank.
- Two pairs: Two cards of the same rank, and two other cards of another same rank.
- One pair: Just a pair of cards (two cards) of the same rank.
- High card: Any other hand.

For the purpose of a sequence, J, Q and K are treated as 11, 12 and 13 respectively. In a straight (or a straight flush), A can be seen as a rank either above K or below 2, thus both A-K-Q-J-10 and 5-4-3-2-A are possible (but not 3-2-A-K-Q or the likes).

If more than one player has the same kind of hand, ties are broken by comparing the ranks of the cards. The basic idea is to compare first those forming sets (pairs, triples or quads) then the rest cards one by one from the highest-ranked to the lowest-ranked, until ties are broken. More specifically:

- Royal straight flush: (ties are not broken)
- Straight flush: Compare the highest-ranked card.
- Four of a kind: Compare the four cards, then the remaining one.
- Full house: Compare the three cards, then the pair.
- Flush: Compare all cards one by one.
- Straight: Compare the highest-ranked card.
- Three of a kind: Compare the three cards, then the remaining two.
- Two pairs: Compare the higher-ranked pair, then the lower-ranked, then the last one.
- One pair: Compare the pair, then the remaining three.
- High card: Compare all cards one by one.

The order of the ranks is A, K, Q, J, 10, 9, ..., 2, from the highest to the lowest, except for the straight (or straight flush) 5-4-3-2-A, where A is considered lower than 2. Note that there are exceptional cases where ties remain. Also, note that the suits are not considered at all in tie-breaking.

Now, given YP's and PY's hole cards and the community cards, determine who wins the game or the game is tied.

Input

The first line of the input contains an integer T denoting the number of testcases in this input file.

For each testcase, it contains three lines. The first line contains four strings denoting YP's hole cards. The second line contains four strings denoting PY's hole cards. The third line contains five strings denoting the community cards.

Each card is represented by two characters. The first one indicates the suit: `S` (spades), `H` (hearts), `D` (diamonds), or `C` (clubs). The second one indicates the rank: `A`, `K`, `Q`, `J`, `T` (10), or `9`–`2`.

- $1 \leq T \leq 5 \times 10^4$
- In a single testcase, all cards are distinct.

Output

For each testcase, output a string in a single line.

If YP wins the game, output `"YP win"` without quotes.

If PY wins the game, output `"PY win"` without quotes.

If the game is tie, output `"Tie"` without quotes.

Sample Input 1

```
3
CA S2 D3 H4
SA C2 H3 D4
S8 SK DT DJ CQ
CA S2 D3 H4
S5 C2 H3 D4
S8 SK DT DJ CQ
C5 S2 D3 H4
SA C2 H3 D4
S8 SK DT DJ CQ
```

Sample Output 1

```
Tie
YP win
PY win
```

This page is intentionally left blank.

L. Loli

Problem ID: loli

YP is a lolicon, he wants to chat to as many lolis as possible.

YP downloads a loli-app, where he can chat with lolis online. There are N lolis in the app, numbered from 1 to N.

YP buys M tickets. The i-th ticket has a number a_i written on it, which means that he can use the i-th ticket to chat with the loli number a_i.

Additionally, there are K ticket exchange chances that YP can use. In the i-th ticket exchange, YP can use a ticket with the number x_i to exchange a ticket with the number y_i. YP can use each exchange chance **at most once**, but he can choose the order to use the ticket exchange chances. Of course, YP can decide not to use those chances as well.

YP wants to chat with as many different lolis as possible. What's the maximum number of distinct lolis that YP can chat with? Write a program to solve that!

Input

The first line of the input contains three integers N, M, K, denoting the number of lolis in the app, the number of tickets YP buys, and the number of ticket exchange chances.

The second line contains M integers $a_1, a_2, \ldots, a_M$, a_i means the number written on the i-th card.

In the next K lines, the i-th line contains two integers x_i, y_i. The meaning of those variables has been mentioned above.

- $1 \leq N, M \leq 50$
- $0 \leq K \leq 50$
- $1 \leq a_i, x_i, y_i \leq N$

Output

Output one number in one line. The number means that maximum number of distinct lolis that YP can chat with.

Sample Input 1	Sample Output 1
2 2 0 1 1	1

Sample Input 2	Sample Output 2
4 3 1 1 2 2 2 4	3