Problem A. Apple Business

Chitanda owns a farm with n apple trees on it, labeled by $1, 2, \ldots, n$, the i-th apple tree has a_i apples. There are $n-1$ bidirectional roads connecting them. For all $i > 1$, an apple tree number i is connected by a road to the apple tree number $\lfloor \frac{i}{2} \rfloor$. So the map of the farm is also like a tree.

Apple business is booming. There are m requests, labeled by $1, 2, \ldots, m$. For the i-th request, Chitanda can sell the customer at most c_i apples, and the customer will pay w_i dollars for each apple. Unfortunately, different customers have different preferences. Specifically, for the i-th request, the customer will give two integers u_i and v_i, denoting he only accepts apples from trees on the shortest path from the u_i-th apple tree to the v_i-th apple tree on the map(include u_i and v_i).

Assume the root of the map is the 1-th apple tree. It is amazing that u_i is always v_i's ancestor or $u_i = v_i$.

The farm may not be able to sell c_i apples to each customer. Please write a program to help Chitanda sell apples that will maximize his profits.

Input

The first line of the input contains an integer $T (1 \le T \le 1000)$, denoting the number of test cases.

In each test case, there are two integers $n, m (1 \le n, m \le 100000)$ in the first line, denoting the number of apple trees and requests.

In the second line, there are n integers $a_1, a_2, ..., a_n (1 \le a_i \le 10^9)$, denoting the number of apples on each apple tree.

For the next m lines, each line contains four integers $u_i, v_i, c_i, w_i (1 \le u_i, v_i \le n, 1 \le c_i \le 10^9, 1 \le w_i \le 10^4)$, denoting each request. It is guaranteed that u_i is v_i's ancestor or $u_i = v_i$.

It is guaranteed that $\sum n \le 10^6$ and $\sum m \le 10^6$.

Output

For each test case, print a single line containing an integer, denoting the maximum profits.

Example

standard input	standard output
1 5 3 2 1 3 1 1 2 5 2 3 2 4 2 4 1 2 3 1	13

Problem B. Balanced Diet

Taylor is wandering in a milk candy store. The store has m types of sweets and there are n sweets in the store. The i-th sweet has the value of a_i, and it is of type b_i.

Taylor is planning to buy some sweets in the store, each sweet can be bought at most once. He will buy at least one sweet. Taylor knows that a balanced diet is important, the value of a sweet set is measured as $\frac{S}{C}$, where S denotes the sum of a_i and C denotes the maximum number of occurrences among all types of sweets.

Assume Taylor selects p_i sweets of type i, it is not welcomed if $1 \leq p_i < l_i$. Note that p_i can also be 0 and p_i can be everything when $l_i = 1$.

Please write a program to help Taylor find the sweet set with maximum value.

Input

The first line of the input contains an integer $T(1 \leq T \leq 1000)$, denoting the number of test cases.

In each test case, there are two integers $n, m(1 \leq n, m \leq 100000)$ in the first line, denoting the number of sweets and types.

In the second line, there are m integers $l_1, l_2, ..., l_m (1 \leq l_i \leq n)$.

For the next n lines, each line contains two integers $a_i, b_i (1 \leq a_i \leq 10^8, 1 \leq b_i \leq m)$, denoting each sweet.

It is guaranteed that $\sum n \leq 10^6$ and $\sum m \leq 10^6$, and there always exists a valid sweet set.

Output

For each test case, print a single line of format `u/v`, denoting the maximum value $\frac{u}{v}$. Note that you should guarantee that $\gcd(u, v) = 1$.

Example

standard input	standard output
2 2 1 2 7 1 2 1 3 2 1 2 2 1 5 2 3 2	9/2 5/1

Problem C. Line-line Intersection

There are n lines $l_1, l_2, \ldots, l_n$ on the 2D-plane.

Staring at these lines, Calabash is wondering how many pairs of (i, j) that $1 \le i < j \le n$ and l_i, l_j share at least one common point. Note that two overlapping lines also share common points.

Please write a program to solve Calabash's problem.

Input

The first line of the input contains an integer $T(1 \le T \le 1000)$, denoting the number of test cases.

In each test case, there is one integer $n(1 \le n \le 100000)$ in the first line, denoting the number of lines.

For the next n lines, each line contains four integers $xa_i, ya_i, xb_i, yb_i(|xa_i|, |ya_i|, |xb_i|, |yb_i| \le 10^9)$. It means l_i passes both (xa_i, ya_i) and (xb_i, yb_i). (xa_i, ya_i) will never be coincided with (xb_i, yb_i).

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

Output

For each test case, print a single line containing an integer, denoting the answer.

Example

standard input	standard output
3 2 0 0 1 1 0 1 1 0 2 0 0 0 1 1 0 1 1 2 0 0 1 1 0 0 1 1	1 0 1

Problem D. Master of Data Structure

Professor Elephant is a master of data structure. Recently, he invented a perfect data structure to maintain information on a tree. He is very proud of this invention, so he prepared this problem for you.

You are given a tree with n nodes. The tree nodes are numbered from 1 to n. The i-th node has an integer value w_i.

Initially, $w_i = 0$ for all $i \in [1, n]$.

Let's denote $p(u, v)$ as all nodes x on the shortest path from the u-th node to the v-th node(include u and v). There will be m events of 7 kinds below:

- `1 u v k` ($1 \leq u, v \leq n, 1 \leq k \leq 10^5$), for all nodes $x \in p(u, v)$, change w_x to $w_x + k$.
- `2 u v k` ($1 \leq u, v \leq n, 1 \leq k \leq 10^8$), for all nodes $x \in p(u, v)$, change w_x to $w_x \oplus k$, where "$\oplus$" denotes the bitwise XOR operation.
- `3 u v k` ($1 \leq u, v \leq n, 1 \leq k \leq 10^8$), for all nodes $x \in p(u, v)$, change w_x to $w_x - k$. Note that if $w_x < k$, ignore such x.
- `4 u v` ($1 \leq u, v \leq n$), ask for the sum of w_x where $x \in p(u, v)$.
- `5 u v` ($1 \leq u, v \leq n$), ask for the bitwise XOR sum of w_x where $x \in p(u, v)$.
- `6 u v` ($1 \leq u, v \leq n$), ask for $\max\{w_x | x \in p(u, v)\} - \min\{w_x | x \in p(u, v)\}$.
- `7 u v k` ($1 \leq u, v \leq n, 1 \leq k \leq 10^8$), ask for $\min\{|w_x - k| | x \in p(u, v)\}$.

Please write a program to support these events efficiently.

Input

The first line of the input contains an integer $T (1 \leq T \leq 5)$, denoting the number of test cases.

In each test case, there are two integers $n, m (1 \leq n \leq 500000, 1 \leq m \leq 2000)$ in the first line, denoting the number of nodes and events.

For the next $n - 1$ lines, each line contains two integers u and v, denoting a bidirectional edge between vertex u and v.

For the next m lines, each line describes an event.

Output

For each query event, print a single line containing an integer, denoting the answer.

Example

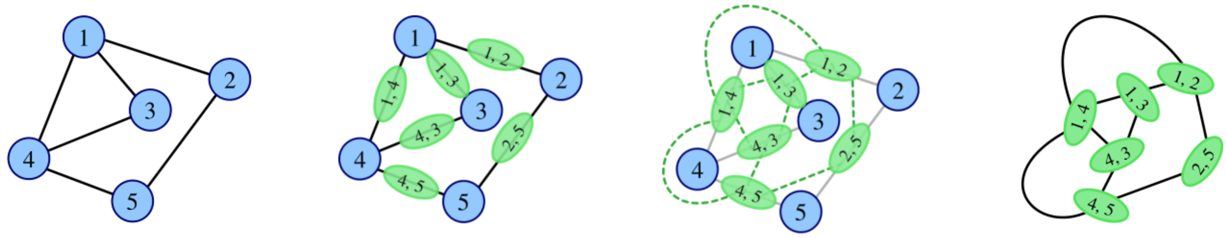
standard input	standard output
1 5 8 5 2 5 1 2 4 2 3 1 4 4 5 3 4 4 1 2 3 1 4 6 3 5 4 2 5 5 1 3 6 5 4 7 1 4 2	0 8 0 0 2

Problem E. Minimum Spanning Tree

In the mathematical discipline of graph theory, the line graph of a simple undirected weighted graph G is another simple undirected weighted graph $L(G)$ that represents the adjacency between every two edges in G.

Precisely speaking, for an undirected weighted graph G without loops or multiple edges, its line graph $L(G)$ is a graph such that:

- Each vertex of $L(G)$ represents an edge of G.
- Two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G, and the weight of such edge between this two vertices is the sum of their corresponding edges' weight.



A minimum spanning tree(MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible.

Given a tree G, please write a program to find the minimum spanning tree of $L(G)$.

Input

The first line of the input contains an integer $T(1 \le T \le 1000)$, denoting the number of test cases.

In each test case, there is one integer $n(2 \le n \le 100000)$ in the first line, denoting the number of vertices of G.

For the next $n-1$ lines, each line contains three integers $u, v, w(1 \le u, v \le n, u \ne v, 1 \le w \le 10^9)$, denoting a bidirectional edge between vertex u and v with weight w.

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

Output

For each test case, print a single line containing an integer, denoting the sum of all the edges' weight of $MST(L(G))$.

Example

standard input	standard output
2 4 1 2 1 2 3 2 3 4 3 4 1 2 1 1 3 1 1 4 1	8 4

Problem F. Mini-game Before Contest

On the day before this contest, two CCPC teams are playing a game after dress rehearsal.

As we know, a CCPC team consists of 3 contestants, so there are 6 players in the game. The game is played on a directed graph. The graph has n vertices and m arcs. These vertices are labeled by $1, 2, \ldots, n$.

Initially, there is a token placed in one of the graph vertices. Players take turns moving the token along one of the arcs that starts in the vertex the token is currently in. When there is no such arc, then this player's team loses the game.

Let's mark two teams as A and B, then the order of 6 players can be displayed as a string ord with length 6. Player from team ord_1 moves first, then $ord_2, ord_3, \ldots, ord_6, ord_1 \ldots$

Each player will follow a strategy that leads his team to win, and if there is no possible to win, he will try to make the game run infinitely. But some contestants announce they are "actors" before the game, each of them will follow a strategy that leads his team to lose, and if there is no possible to lose, he will try to make the game run infinitely.

All the players know what the order is and who are actors. They also know preferences of other players. They will play optimally, but they are not allowed to discuss before or during the game.

For each initial position of the token, your task is to determine what kind of result the game is going to have.

Input

The first line of the input contains an integer $T (1 \le T \le 100)$, denoting the number of test cases.

In each test case, there is two integers $n (1 \le n \le 100000, 1 \le m \le 200000)$ in the first line, denoting the number of vertices and arcs.

For the next m lines, each line contains two integers $u_i, v_i (1 \le u_i, v_i \le n)$, denoting an arc from vertex u_i to vertex v_i. Note that u_i can be the same with v_i, but no arc will appear more than once.

The next line contains a string ord of length 6, denoting the order. There are exactly 3 of which are "`A`" and the others are "`B`".

The next line contains six integers $a_1, a_2, \ldots, a_6 (0 \le a_i \le 1)$, where a_i denotes whether the player corresponding to ord_i is an actor. Specifically, $a_i = 1$ means he is an actor while $a_i = 0$ means not.

It is guaranteed that $\sum n \le 2 \times 10^6$ and $\sum m \le 3 \times 10^6$.

Output

For each test case, print a single line containing n characters, where the i-th character should denote the result of the game if the token is in vertex i initially. The result of the game is denoted by "`A`" if the team A wins the game, "`B`" if the team B wins the game, and "`D`" if the game runs infinitely.

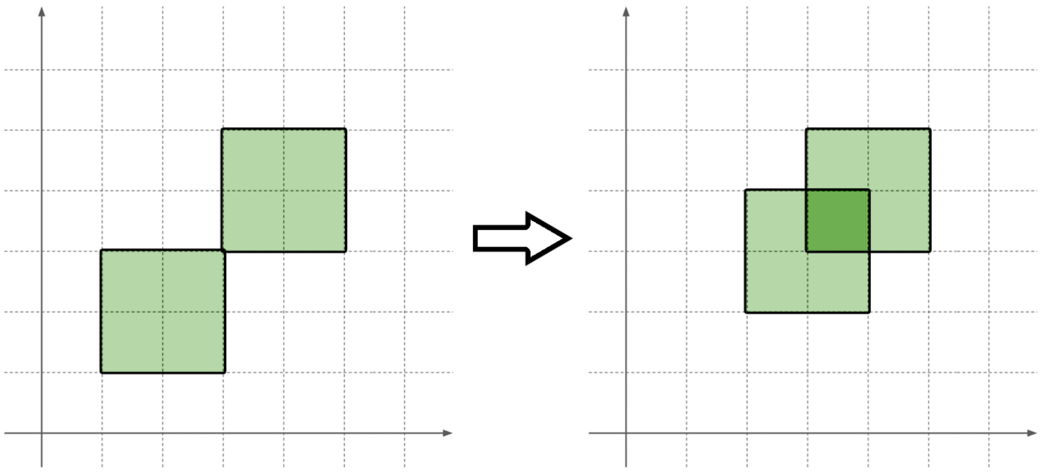
Example

standard input	standard output
2 4 4 1 2 2 3 3 1 1 4 AAABBB 000000 6 6 1 2 2 3 3 1 1 4 2 5 5 6 AAABBB 001000	DADB ABBBBB

Problem G. Radar Scanner

There are n rectangle radar scanners on the ground. The sides of them are all paralleled to the axes. The i-th scanner's bottom left corner is square (a_i, b_i) and its top right corner is square (c_i, d_i). Each scanner covers some squares on the ground.

You can move these scanners for many times. In each step, you can choose a scanner and move it one square to the left, right, upward or downward.



Today, the radar system is facing a critical low-power problem. You need to move these scanners such that there exists a square covered by all scanners.

Your task is to minimize the number of move operations to achieve the goal.

Input

The first line of the input contains an integer $T (1 \leq T \leq 1000)$, denoting the number of test cases.

In each test case, there is one integer $n (1 \leq n \leq 100000)$ in the first line, denoting the number of radar scanners.

For the next n lines, each line contains four integers $a_i, b_i, c_i, d_i (1 \leq a_i, b_i, c_i, d_i \leq 10^9, a_i \leq c_i, b_i \leq d_i)$, denoting each radar scanner.

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

Output

For each test case, print a single line containing an integer, denoting the minimum number of steps.

Example

standard input	standard output
1 2 2 2 3 3 4 4 5 5	2

Problem H. Skyscraper

At the main street of Byteland, there will be built n skyscrapers, standing sequentially one next to other. If look leftside right, sequence of their height will be $a_1, a_2, \dots, a_n$.

Initially the street is empty, every skyscraper's height is 0. Hamster is the leader of the construction team. In each stage, Hamster can select a range $[l, r]$, then the team will work on this range. Specifically, assume the height sequence is $h_1, h_2, \dots, h_n$, then $h_l, h_{l+1}, \dots, h_r$ will increase by 1 during this stage. When $h_i = a_i$ holds for all $i \in [1, n]$, the project will be closed.

The plan may be changed for many times. There will be m events of 2 kinds below:

- `1 l r k` $(1 \le l \le r \le n, 1 \le k \le 10^5)$, for all $x \in [l, r]$, change a_x to $a_x + k$.
- `2 l r` $(1 \le l \le r \le n)$, assume $a_1, a_2, \dots, a_{l-1}, a_{r+1}, a_{r+2}, \dots, a_n = 0$, ask for the minimum number of required stages to close the project.

Input

The first line of the input contains an integer $T(1 \le T \le 1000)$, denoting the number of test cases.

In each test case, there are two integers $n, m(1 \le n, m \le 100000)$ in the first line, denoting the number of skyscrapers and events.

In the second line, there are n integers $a_1, a_2, ..., a_n(1 \le a_i \le 100000)$.

For the next m lines, each line describes an event.

It is guaranteed that $\sum n \le 10^6$ and $\sum m \le 10^6$.

Output

For each query event, print a single line containing an integer, denoting the answer.

Example

standard input	standard output
1 5 4 1 3 1 4 5 2 1 5 1 3 4 2 2 2 4 2 1 5	7 6 6

Problem I. Temperature Survey

Doctor Sunset is doing a survey about global warming. He selected two cities A and B, and has gathered the average air temperature of these two cities throughout passed n days. On the i-th day, the average air temperature of A was a_i while the average air temperature of B was b_i.

Doctor Sunset drew a key graph using the temperature data. From his observation, there are several facts about the data:

- All the values are integers within $[1, n]$.
- Global warming is ture, so the temperature is non-decreasing, which means $a_1 \le a_2 \le a_3 \le \cdots \le a_n$ and $b_1 \le b_2 \le b_3 \le \cdots \le b_n$.
- City B is always not hotter than A, which means $b_i \le a_i$ for all $i \in [1, n]$.

Unfortunately, Doctor Sunset erased all the data of city B by mistake just now. But he can recover the array b by the facts described above. Please write a program to help Sunset count the number of possible arrays that can be b.

Input

The first line of the input contains an integer $T (1 \le T \le 100)$, denoting the number of test cases.

In each test case, there is one integer $n (1 \le n \le 2 \times 10^5)$ in the first line, denoting the number of days.

In the second line, there are n integers $a_1, a_2, ..., a_n (1 \le a_i \le n)$, denoting the average air temperature of city A. The input is always valid, so you can assume $a_1 \le a_2 \le a_3 \le \cdots \le a_n$.

It is guaranteed that $\sum n \le 5 \times 10^5$.

Output

For each test case, print a single line containing an integer, denoting the number of possible arrays that can be b. As the answer can be very large, output it modulo 998244353.

Example

standard input	standard output
3 4 1 1 1 1 4 1 2 3 4 4 4 4 4 4	1 14 35

Problem J. Time Limit

In CCPC contests, you will get "Time Limit Exceeded" when your program tried to run during too much time. Setting suitable time limit for problems is vital to a contest.

Mr. Bread is preparing problems for a coming contest with his friends. For each problem, there will be a "Main Correct Solution" denotes the standard solution program written by the author. There will also be several "Correct Solutions" denote solution programs intended to pass.

Assume there are n programs in total, labeled by $1, 2, \ldots, n$. The 1-th program denotes the "Main Correct Solution" while others are "Correct Solutions". The i-th program runs in a_i seconds.

According to the rules in Mr. Bread's mind, the time limit x should meet all the rules below:

- The constraint can't be too tight, which means $x \geq 3a_1$.
- All the "Correct Solutions" should pass, which means $x \geq a_i + 1$ for all $i \in [2, n]$.
- x should be the smallest even integer meeting the rules described above.

Please write a program to find the time limit x.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10)$, denoting the number of test cases.

In each test case, there is one integer $n(2 \leq n \leq 10)$ in the first line, denoting the number of programs.

In the second line, there are n integers $a_1, a_2, ..., a_n(1 \leq a_i \leq 10)$.

Output

For each test case, print a single line containing an integer, denoting the value of x.

Example

standard input	standard output
2 2 1 3 2 1 4	4 6