2021 Jiangsu Collegiate Programming Contest

December 25

JSCPC Jiangsu Collegiate Programming Contest

Problems

A. Spring Couplets
B. Among Us
C. Magical Rearrangement
D. Pattern Lock
E. Stone Ocean
F. Jumping Monkey II
G. Five Phases
H. Reverse the String
I. Fake Walsh Transform
J. Anti-merge
K. Longest Continuous 1
L. Tree Game

Do not open before the contest starts.

Problem A. Spring Couplets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Spring Festival 2022 is coming. Vingying can't hold back his excitement and starts to prepare the spring couplets.

A spring couplet is a pair of lines of poetry that adhere to specific rules. They can be seen on doorways in Chinese communities, displayed as part of the Spring Festival. The text of the couplets is often traditional and contains hopes for prosperity.

A spring couplet needs to meet various rules. For example, both lines must have the same number of Chinese characters, and the meanings of the two lines must be related, and so on. Here, we only focus on the **tonal rules**.

As we know, there are four primary tones of Chinese characters: flat ($\bar{a}$), rising ($\acute{a}$), falling-rising ($\check{a}$), and falling ($\grave{a}$). These can be categorized into two tone patterns: *level tones* and *oblique tones*. In detail, the flat tone and rising tone belong to *level tones*, and the falling-rising tone and falling tone belong to *oblique tones*.

A spring couplet must adhere to the following **tonal rules**:

- The tone pattern of one line must be the inverse of the other. It means if one character is of a *level tone*, its corresponding character in the other line must be of an *oblique tone*, and vice versa.
- The last character of the first line should be of an *oblique tone*, which forces the last character of the second line to be of a *level tone*.

Due to Vingying's creative mind, he creates T couplets instantly. He wants to know whether each couplet adheres to the **tonal rules**.

Input

The first line of the input contains an integer T ($1 \leq T \leq 100$), indicating the number of couplets.

For each couplet, the first line contains an integer n ($1 \leq n \leq 20$), indicating the number of Chinese characters for each line.

For the next two lines, each line contains n strings, each of which represents a Chinese character. Each string consists of lowercase letters, indicating the spelling of the Chinese character, followed by a digital number 1 (flat), 2 (rising), 3 (falling-rising), or 4 (falling) representing the tone.

It is guaranteed that the length of each string is no more than 7 and no less than 2.

Output

For each couplet, if it adheres to the **tonal rules**, print `YES` in a single line. Otherwise, print `NO` in a single line.

Example

standard input	standard output
4 7 qian1 men2 wan3 hu3 tong2 tong2 ri4 zong3 ba3 xin1 tao2 huan4 jiu4 fu2 7 ping2 ping2 ze4 ze4 ping2 ping2 ze4 ze4 ze4 ping2 ping2 ze4 ze4 ping2 7 gou3 li4 guo2 jia1 sheng1 si3 yi3 qi3 yin1 huo4 fu2 bi4 qu1 zhi1 3 nun1 heh1 heh1 a4 a4 a4	YES YES NO NO

Note

For the third example, the tones of the first character do not adhere to the rule, as they are both *oblique tones*.

For the fourth example, the tones of the last character do not adhere to the rule, as the last character in the first line is not of an *oblique tone*.

Problem B. Among Us

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Among Us is an online multiplayer social deduction game developed and published by American game studio Innersloth. Dropped into a spaceship, each player is designated as a private role of either a *crewmate* or an *impostor*. In this problem, there are exactly 2 imposters and at most 8 crewmates. The imposters need to kill all the crewmates to win the game, while the crewmates need to complete the tasks as quickly as possible.

There are n rooms in the game map. When the game begins, each player will spawn in a certain room. Players have two choices for each second: stand still or move to another room. There are m undirected secret paths for imposters to move, each of which connects two rooms and takes a certain amount of time to pass through. In this problem, the crewmates can move freely among the rooms, while the imposters can only use the m secret paths.

As the imposters are extremely smart, they can predict where some crewmates will be at some moments. When a crewmate appears in a room as predicted and at least one imposter is there, the imposter can kill the crewmate without consuming time. The imposters also predict that the crewmates will finish all the tasks in $t_{\max}$ seconds. That is, if at least one crewmate survives after $t_{\max}$ seconds, the imposters will lose the game. Based on the prediction, please calculate the minimum time for the imposters to kill all the crewmates if possible.

Input

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

The first line of each test case contains three integers n, m, k ($1 \le n \le 10^4$, $1 \le m \le 2 \times 10^4$, $1 \le k \le 8$), denoting the number of rooms, secret paths, and crewmates respectively.

Each of the next m lines contains three integers u, v, w ($1 \le u, v \le n$, $u \ne v$, $1 \le w \le 10^4$), denoting there is a secret path connecting room u and v, and imposters need exactly w seconds to pass through it.

The next line contains two integers e and $t_{\max}$ ($1 \le e \le 10^5$, $1 \le t_{\max} \le 10^8$), denoting the number of predictions and the time for the crewmates to complete all the tasks.

Each of the next e lines contains three integers p, x, t ($1 \le p \le k$, $1 \le x \le n$, $1 \le t \le t_{\max}$), denoting the crewmate p will appear in room x at t seconds after the game begins.

The last line of each test case contains two integers x and y ($1 \le x, y \le n$), denoting that the two imposters will spawn in room x and y when the game begins.

It's guaranteed that $\sum n \le 10^4$, $\sum m \le 2 \times 10^4$ and $\sum e \le 10^5$ over all test cases.

Output

For each test case, output the minimum time for the imposters to kill all the crewmates in a single line. If the imposters can't win the game, output -1 in a single line.

Example

standard input	standard output
4 5 7 3 1 2 1 2 3 2 2 4 1 3 4 5 4 5 1 5 2 5 5 1 5 3 8 1 5 3 3 4 2 2 2 4 3 1 3 1 2 1 2 1 3 3 1 2 1 2 2 1 1 2 2 2 3 3 1 2 1 2 1 3 3 1 2 1 2 2 1 1 2 2 3 3 10 10 8 1 2 1 2 3 1 3 4 1 4 5 1 5 6 1 6 7 1 7 8 1 8 9 1 9 10 1 10 1 1 8 10 1 2 1 2 3 2 3 4 3 4 5 4 5 7 1 6 8 2 7 9 3 8 10 4 1 6	4 1 -1 4

Problem C. Magical Rearrangement

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

HoshiYo is a magician. He specializes in using magic, but he is not good at math. In the math class of the magic school, HoshiYo learns about integers. He suddenly finds something interesting: with his powerful magic, he can change an integer by rearranging its digits.

Formally, for each digit from 0 to 9, the number of digits i is a_i. HoshiYo wants to get an integer that meets the following rules:

- All the given digits are used.
- There is no leading zero except for 0 itself.
- Adjacent digits cannot be the same.

HoshiYo wonders what's the minimum integer he can get with these digits.

Input

The first line of each test case contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases.

Each test case contains 10 integers $a_0, a_1, \ldots, a_9$ ($0 \le a_i \le 10^5$), indicating the number of different digits. Let $n = \sum_{i=0}^{9} a_i$. It's guaranteed that $1 \le n \le 10^5$.

It's also guaranteed that the sum of n over all test cases will not exceed 10^5.

Output

For each test case, output the minimum integer HoshiYo can get in a single line. If there's no solution, output -1 in a single line.

Example

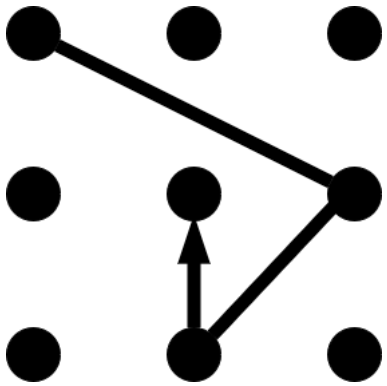
standard input	standard output
3 0 0 1 0 0 0 0 0 0 2 2 0 1 0 0 1 0 2 0 0 3 0 1 0 0 0 0 0 0 0	929 205707 -1

Problem D. Pattern Lock

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Usually, smartphones can be locked by either a password, fingerprint sensing, or facial recognition. There's also another commonly used method: *Pattern Lock*.

Pattern Lock allows you to lock and unlock your device by drawing a pattern onto the screen. The pattern is drawn by connecting a series of dots on a grid by a polygonal path. The grid of dots on the lock screen has n rows and m columns. The rows have equal spacing between each other, and so do the columns. We denote the dots in the x-th row and y-th column by (x, y). And we use a sequence of dots to denote the pattern, i.e. the polygonal path. For example, the sequence $\{(1,1),(2,3),(3,2),(2,2)\}$ can denote the pattern shown in the picture below.



Let's denote a pattern with k dots as $\{A_1, A_2, \ldots, A_k\}$. A valid pattern to lock a smartphone should meet the following conditions:

- Each dot is visited no more than once. That is, for each $1 \le i < j \le k$, $A_i \neq A_j$.
- For each $1 \le i < k$, the segment connecting A_i and A_{i+1} cannot pass through other dots. For example, $A_i = (1,1), A_{i+1} = (3,3)$ is invalid because the segment passes through $(2,2)$.

Little Rabbit wants his smartphone to be as secure as possible. Therefore, he needs a **strong pattern** to lock his device. A **strong pattern** is a valid pattern that meets some extra conditions:

- Each dot is visited exactly once. That is, $k = n \times m$.
- For each $1 < i < k$, the angle formed by segment A_iA_{i-1} and segment A_iA_{i+1} must be an acute angle (less than 90°).

Can you construct a **strong pattern** for him?

Input

The input contains two integers n and m ($2 \le n, m \le 500$), representing the number of rows and columns of the grid.

Output

Output $n \times m$ lines. The i-th line contains two integers x_i and y_i ($1 \le x_i \le n$, $1 \le y_i \le m$), representing the i-th dot of the pattern is (x_i, y_i).

It can be proved that the answer always exists. If there are multiple answers, output any.

Example

standard input	standard output
2 2	1 1 2 1 1 2 2 2

Note

Please note that if the length or format of your output does not match the answer, you will possibly get a Presentation Error verdict.

Problem E. Stone Ocean

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4.5 seconds
Memory limit:	512 megabytes

In *JoJo*'s world, some people are capable of transforming their inner spiritual power into a *Stand*. Cujoh Jolyne, like his father, has *Stand Power*. Her Stand is called *Stone Free* which can manipulate strings. Unfortunately, she was framed and sentenced to 15 years in the Green Dolphin Prison. She needs to use her *Stand Power* to help her regain her freedom from the Stone Ocean.

Since she has just acquired her *Stand Power*, it will take her some time to get used to it. Now there are n strings $S_1, S_2, \ldots S_n$. She wants to train her power with these strings by the following steps:

1. Set index $i = 1$, and T as an empty string.
2. Choose a character from S_i uniformly and randomly, which means the probability of each character being selected is $\frac{1}{|S_i|}$, where $|S_i|$ is the length of S_i.
3. Append the chosen character to the back of T.
4. If $i < n$, increase i by 1 and go back to step 2.

After these steps, Cujoh Jolyne gets another string T. She defines the power value of T as the number of pemutations $p_1, p_2, \ldots, p_n$ that satisfy the following condition: $T_{p_1}T_{p_2}....T_{p_n}$ is a palindrome.

Recall that a palindrome is defined as a string that is identical when read from left to right or right to left. For example, `aa`,`aba`,`acca` are palindromes while `ab`,`cab` are not. A permutation $p_1, p_2, \ldots, p_n$ is a sequence where every integer from 1 to n appears exactly once.

To estimate the strength of her power, Cujoh Jolyne wants to know the expectation of the power value of string T.

Input

The first line contains an integer n ($2 \le n \le 30$).

For the next n lines, the i-th line contains a string S_i ($1 \le |S_i| \le 50000$, where $|S_i|$ is the length of S_i) consisting of lowercase letters only.

Output

Output an integer indicating the expectation of the power value of string T modulo 998244353. Formally, let $M = 998244353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \neq 0$. Output $p \cdot q^{-1} \bmod M$, where q^{-1} denotes the multiplicative inverse of q modulo M.

Examples

standard input	standard output
2 ab ac	499122177
4 aabcc abab bbaa acac	399297744

Note

For the first example, string T can be `aa`, `ac`, `ba`, `bc` with the same probability, and the power values of them are $2, 0, 0, 0$ respectively. So the expectation of the power value is $\frac{2+0+0+0}{4} = \frac{1}{2}$.

Problem F. Jumping Monkey II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

There is a tree with n nodes and $n-1$ edges that make all nodes connected. Each node i has a weight a_i. A monkey is jumping on the tree. In each jump, he can go from one node to another through an edge. If the monkey starts from node u_1, passes through $u_2, u_3, \ldots, u_{k-1}$, and ends in u_k, where all visited nodes are distinct, we call $u_1, u_2, u_3, \ldots, u_k$ a simple path. Let's define the *LIS* (Longest Increasing Subsequence) of a simple path $u_1, u_2, u_3, \ldots, u_k$ as follow:

- The longest sequence $p_1, p_2, \ldots, p_l$ that meets $1 = p_1 < p_2 < \cdots < p_l \leq k$ and $a_{u_{p_1}} < a_{u_{p_2}} < \cdots < a_{u_{p_l}}$.

The monkey wants to make the *LIS* as long as possible. For each $i \in [1, n]$, please calculate the maximum length of the *LIS* he can get if he starts from node i.

Input

The first line of the input contains an integer T ($1 \leq T \leq 10^4$), indicating the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 2 \times 10^5$), indicating the number of nodes.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$), indicating the weight of each node.

Each of the following $n-1$ lines contains two integers u, v ($1 \leq u, v \leq n$), indicating an edge connecting nodes u and v. It is guaranteed that the given edges form a tree.

It is guaranteed that $\sum n \leq 2 \times 10^5$ over all test cases.

Output

For each test case, output n lines, the i-th of which is the maximum length of the *LIS* the monkey can get if he starts from node i.

Example

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

Problem G. Five Phases

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

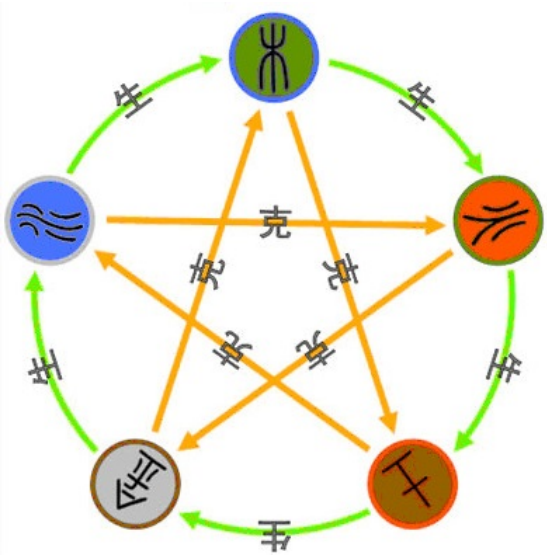
The *Five Phases* is a fivefold conceptual scheme that many traditional Chinese fields used to explain a wide array of phenomena, namely Fire, Water, Wood, Metal, and Earth. There are two kinds of basic interactions between them, each forming a cycle.

The *generating cycle* is

- Wood → Fire → Earth → Metal → Water → Wood

And the *overcoming cycle* is

- Wood → Earth → Water → Fire → Metal → Wood



Yukikaze the Great Mage is designing magic based on the five phases. By embedding intricate geometric patterns, she can adjust the amount of each phase imbued to the magic. We use five integral parameters, the amount of each phase imbued, to describe the magic invented by Yukikaze.

There are four types of perturbation methods to modify the parameters of the magic in the way she wants.

Single Phase Tweaking Increase or decrease the amount of one phase by one.

Positive Propagation Tweaking Increase the amount of the phase x by one. Then if we have x generates y in and y generates z, the amount of y and z is also increased by one. Or decrease the amount of the phase x by one. Then if we have x generates y in and y generates z, the amount of y and z is also decreased by one.

Negative Propagation Tweaking Increase the amount of the phase x by one. Then if we have x generates y and y overcomes z, the amount of y is increased by one and the amount of z is decreased by one. Or decrease the amount of the phase x by one. Then if we have x generates y and y overcomes z, the amount of y is decreased by one and the amount of z is increased by one.

Global Tweaking Increase or decrease the amount of all phases by one.

For example, we have Metal generating Water and Water generating Wood. By applying Positive Propagation Tweaking started by increasing the amount of Metal by one, the result is that the amount of Metal, Water, Wood are all increased by one. For another example, we have Fire generating Earth and Earth overcoming Water. By applying Negative Propagation Tweaking started by decreasing the amount of Fire by one, the result is that the amount of Fire and Earth are both decreased by one, and the amount of water is increased by one.

Yukikaze wants to create some novel magic. Initially, the five parameters are all zero. Now given the five parameters describing the magic and the number of steps she wants to use, she wonders how many ways there are to create the magic by applying the perturbation methods described above, exactly one perturbation method in each step. Two ways are considered different if there exists an integer i that methods used in the i-th step are different. Since the answer can be too large, output the answer modulo 998244353.

Input

The first line contains a single integer T ($1 \le T \le 10^5$), denoting the number of magic that Yukikaze wants to create.

Each of the next T lines contains six integers c_{Wood}, c_{Fire}, c_{Earth}, c_{Metal}, c_{Water} ($-10^5 \le c_{\text{Wood}}, c_{\text{Fire}}, c_{\text{Earth}}, c_{\text{Metal}}, c_{\text{Water}} \le 10^5$) and k ($0 \le k \le 10^5$), indicating the five parameters of the magic and the number of steps Yukikaze wants to use.

Output

For each magic, output the number of ways to create it in a single line, modulo 998244353.

Example

standard input	standard output
5 0 0 0 0 0 1 0 0 0 3 2 5 1 1 1 1 1 5 3 2 1 1 1 4 0 -1 0 2 4 5	0 10000 100000 576 2500

Problem H. Reverse the String

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There is a string of lowercase letters, and you want to minimize its lexicographical order. What you can do is reverse an interval or do nothing.

For example, for the string `abcdefg`, if we reverse the interval `abcdefg`, it will become `abfedcg`.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$.
- In the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

The first line contains an integer T ($1 \leq T \leq 20$), denoting the number of test cases.

For the following T lines, each line contains a string s of lowercase letters ($1 \leq |s| \leq 10^5$, where $|s|$ is the length of s).

It is guaranteed that $\sum |s| \leq 1.5 \times 10^6$ over all test cases.

Output

For each test case, output the string with the minimal lexicographical order in a single line.

Example

standard input	standard output
1 abbcabaac	aaabacbbc

Problem I. Fake Walsh Transform

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There is a set of integer $\{0, 1, 2, \ldots, 2^m - 1\}$. Now you need to select k integers from the set. Each integer should be selected no more than once. Let's denote the integers you selected by $a_1, a_2, \ldots, a_k$. You should make sure that $a_1 \oplus a_2 \oplus \ldots \oplus a_k = n$, where $\oplus$ indicates the bitwise exclusive OR operation.

You want to make k as large as possible. Please calculate the maximum k.

Input

The first line contains one integer T ($1 \le T \le 10^4$), indicating the number of test cases.

For each test case, the only line contains two integers m and n ($1 \le m \le 60$, $0 \le n < 2^m$).

Output

For each test case, output one integer in a single line, indicating the maximum k.

Example

standard input	standard output
1 2 2	3

Note

For example, we can select $\{0, 1, 3\}$ from the set, since $0 \oplus 1 \oplus 3 = 2$.

Problem J. Anti-merge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In Kanade's daily work, there are a lot of documents for her to process. One of Kanade's jobs is to process tables, like this:

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

Kanade doesn't like repetitive numbers. So she installs a plugin to merge adjacent cells with the same content. The plugin will first merge cells with the same content in every **column** and then merge cells with the same content (and height, if ones are merged in the first step) in every **row**. For example, the table above will look like this with the plugin:

<table>
<tr><td rowspan="3">1</td><td rowspan="2">1</td><td rowspan="2" colspan="2">2</td><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>4</td><td>3</td><td colspan="2">5</td></tr>
</table>

This plugin works well for Kanade, but it also poses some problems. For instance, some projects require her not to merge the cells. To meet this, Kanade can add *tags* to some cells. A tag is part of the content of a cell, but it is not displayed. So if two cells display the same number but have different tags, they will not be merged. But if they have the same tags, they will still be merged.

Now Kanade needs to process a table with n rows and m columns which requires her not to merge the cells. To keep any cells from being merged, Kanade would like to know how many kinds of tags at least need to be added. She also wants to know how to add tags to minimize the number of tags when the number of kinds of tags is the least.

Input

The first line of input contains two integers n, m ($1 \le n, m \le 500$), indicating the number of rows and columns of the table.

For the next n lines, each line contains m integers in the range of $[1, 10^9]$, indicating the content of the table.

Output

The first line of the output contains two integers. The first integer t ($0 \le t \le n \times m$) indicates the minimum number of kinds of tags that need to be added. The second integer k ($0 \le k \le n \times m$) indicates the minimum number of tags that need to be added when the number of kinds of tags is the least.

For the next k lines, each line contains three integers x, y, c ($1 \le x \le n$, $1 \le y \le m$, $1 \le c \le t$), indicating that tag c is added to the cell in the x-th row and y-th column. If there are multiple solutions, output any.

Examples

standard input	standard output
1 3 1 1 4	1 1 1 1 1
1 3 5 1 4	0 0
2 3 1 1 4 5 1 4	1 2 1 2 1 2 3 1

Note

Please note that if the length or format of your output does not match the answer, you will possibly get a Presentation Error verdict.

Problem K. Longest Continuous 1

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There is a sequence of binary strings $s_0, s_1, s_2, \ldots$, which can be defined by the following recurrence relation:

- $s_0 = 0$
- $s_i = s_{i-1} + b_i$

Here b_i means the binary form of i without leading zeros. For example, $b_5 = 101$. And $s_{i-1} + b_i$ means to append string b_i to the back of s_{i-1}.

s_0	s_1	s_2	s_3	s_4	s_5	...
0	01	0110	011011	011011100	011011100101	...

Let's use p_k to denote the prefix of $s_{10^{100}}$ of length k. Now given k, please calculate the length of the longest continuous 1 in p_k.

Input

The first line of the input contains one integer T ($1 \le T \le 10^4$), indicating the number of test cases.

For each test case, the only line contains one integer k ($1 \le k \le 10^9$), indicating the length of the prefix.

Output

For each test case, output one integer in a single line, indicating the length of the longest continuous 1 in p_k.

Example

standard input	standard output
4 1 2 3 4	0 1 2 2

Problem L. Tree Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

With great enthusiasm, Alice finds a new game and she is going to play it with Bob.

There's a tree with n nodes rooted at node 1. Each node u has an associated value a_u, where $a_1, a_2, \ldots, a_n$ is a permutation of n (every integer from 1 to n occurs exactly once). Let's define the **rearrangement** operation on a node u as follow:

- Remove all the values on node u and all the children directly connected to u. Then reassign the values to them so that $a_1, a_2, \ldots, a_n$ is still a permutation of n.

After the operation, we call the node u **rearranged**. Alice can **rearrange** a node if and only if it has no directly connected children that are not **rearranged**. What's more, she can **rearrange** a node at most once. If Alice can make $a_u = u$ for every node u, she will win this game.

Alice wants to challenge the game and asks Bob to draw a tree and assign the initial values for every node. Now Bob has already drawn a tree and assigned some values, but before he finishes assigning all the values, he finds that there are some ways of assignment that can make Alice never win the game, which will let Alice down. To avoid this, Bob wants to know how many possible ways of assignment there are that Alice has a chance to win the game. Since the answer can be too large, please output the answer modulo 998244353.

Input

The first line of the input contains an integer T ($1 \le T \le 10000$), representing the number of test cases.

The first line of each test case contains an integer n ($2 \le n \le 200000$), representing the number of nodes.

The second line of each test case contains $n - 1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$), where p_i represents the parent of node i.

The third line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$), representing the initial value of each node. $a_i = 0$ means that node i has not been assigned a value yet. It is guaranteed that all non-zero a_i are distinct.

It is guaranteed that the sum of n over all test cases does not exceed 200000.

Output

For each test case, output an integer in a single line, indicating the number of possible ways of assignment. Since the answer can be too large, please output the answer modulo 998244353.

Example

standard input	standard output
2 7 1 2 2 3 2 4 0 0 0 2 0 0 0 7 1 2 3 3 3 1 0 0 0 0 0 0 0	12 288