A. Harvest Moon

Harvest Moon (牧場物語 Bokujō Monogatari) is a virtual role playing game for the Super Nintendo Entertainment System developed and published by Pack-In-Video (now Marvelous AQL), first released in Japan in 1996, and 1997 in North America. This is the first game in the long-running Harvest Moon titles. A PAL version was released by Nintendo in early 1998 for Western Europe and Oceania, with language localizations for Germany and France.
-- wikipedia

Hard though, it's still an incorrigibly temptation for Moor. In the game, Moor's favorite is to cultivate in his own pasture. His pasture is large - we may consider it as a w×h lattice.
The shop sells A kinds of seeds. The price of the ith seed is Q_i dollars, and it can be sowed to cover a 3×3 lattice. Note that the cover area may contain the grids outside his pasture, but only the grids inside belong to him. Further more, although a grid may be sowed multiple times, the later sowing on this grid exert nothing if the corns are still growing in it.
After ploughing and weeding, the corns get ripe right after N_i days and Moor will go harvest. For instance, if Moor sows the seeds at the beginning of day x, they will get ripe at the beginning of the (x+N_i)th day(or you may consider it as the right end of the (x+N_i-1)th day). After harvest, some of the corns will disappear and the grids they belong to would become empty, while the others may stays and get ripe again and again, in a period of M_i days. For convenience, Moor always leaves the second kind of corns staying, but he will try to keep sowing in those new empty grids. Moor only sows seeds and harvest at the beginning or ending of some day, and you may assume it wastes no time for Moor to make operations.
Initially, Moor's got Y dollars. When selling a 1×1 grid of corns, he will obtain P_i dollars back. Moor wonders the maximum dollars he will get at the end of the Dth day, if he buys only one kind of seeds? Certainly, he may do nothing and keep all dollars in his pocket.

Input

The input contains several test cases.
An integer T(T≤110) will exist in the first line of input, indicating the number of test cases.
Each test case begins with 5 integers w,h,A,D,Y(3≤w,h≤100,0<A,D≤1000,0<Y≤100000).
The following A lines, each with 4 integers Q_i,P_i,N_i,M_i (0<Q_i,P_i≤1000;N_i,M_i≤10000), describe the property of the ith seed. M_i=0 denotes that the seed will not get ripe again after harvest.

Output

Output the maximum dollars he's able to obtain at the end of the Dth day, one per line.

Sample Input

```
1
3 3 2 3 100
100 90 3 0
100 90 2 0
```

Sample Output

```
810
```

B. Letter Tree

A letter tree is a rooted tree that each edge is assigned to a lowercase letter. Node 1 is always considered the root. When making a tour in the tree, one is only allowed to step "down". In other word, if you're now on some node of the tree, you can only make a step to one of its children nodes.
After you travel along a path in the tree, you will obtain a Path String, which is formed by the letters assigned to edges that you just move along. The string exactly records all edges in the order of your visit.
Now we're faced with the problem. Located at some node u on the tree, your task is to move for exactly m steps and obtain a maximal lexicographic Path String. In order to avoid the huge output, you're only required to output the hash code of the string after it is transformed into a 26-base number, where 'a' for 0, 'b' for 1, …, 'z' for 25. For instance, "bac" = $(678)_{26}$ for $678=1\times26^2+0\times26^1+2\times26^0$.

Input

The input contains several test cases.
An integer T(T≤20) will exist in the first line of input, indicating the number of test cases.
Each test case begins with an integer N($2\le N\le10^5$), which denotes the number of nodes in the tree.
The following (N-1) lines describe the edges. An edge is described in the format of (u,v,c), which denotes an undirected edge between u and v with a lowercase letter c assigned.
The following line contains a single integer M($1\le M\le10^5$), indicating the number of queries.
The following M lines, each with a pair of positive integers (u,m), describe the queries. ($1\le m\le10^5$)
The nodes are labeled from 1 to N.

Output

Output the hash code modulo 10^9+7 of the maximal lexicographic Path String for each query, one per line. If it's impossible to move for m steps, output IMPOSSIBLE.

Sample Input

```
1
6
1 2 a
1 3 b
2 4 c
2 5 b
2 6 c
3
1 1
1 2
3 1
```

Sample Output

```
1
2
IMPOSSIBLE
```

C. Partition

Define f(n) as the number of ways to perform n in format of the sum of some positive integers. For instance, when n=4, we have

4=1+1+1+1
4=1+1+2
4=1+2+1
4=2+1+1
4=1+3
4=2+2
4=3+1
4=4

totally 8 ways. Actually, we will have $f(n)=2^{(n-1)}$ after observations.

Given a pair of integers n and k, your task is to figure out how many times that the integer k occurs in such $2^{(n-1)}$ ways. In the example above, number 1 occurs for 12 times, while number 4 only occurs once.

Input

The first line contains a single integer $T(1 \le T \le 10000)$, indicating the number of test cases.

Each test case contains two integers n and $k(1 \le n,k \le 10^9)$.

Output

Output the required answer modulo 10^9+7 for each test case, one per line.

Sample Input

```
2
4 2
5 5
```

Sample Output

```
5
1
```

D. Color the Tree

Alice and Bob are playing games again! This time, they invent a new game called "Color the Tree". They draw a tree with N nodes and, certainly, (N-1) edges connecting them to assure a path between each pair of the nodes. But the tree they play with is a little special - each edge is assigned to a color and a specific value. Initially, the value of each edge is settled while the colors are all white.

When the game starts, Alice and Bob each choose a node as her/his starting node. In each round, Alice firstly makes a move from her current node to another through an edge with a color of white or red, and if the edge is white, she colors it to red. After that, Bob makes a similar move through a white or blue edge from his current node, and if the edge is white, he colors it to blue. The game keeps going on until all the edges are colored to either red or blue.

Alice's goal is to maximize the sum of values of the red edges, and Bob, wants to maximize that of the blue edges.

Given the starting node of them, figure out the maximum sum that Alice is able to obtain if both of them take the best strategy in each round.

Input

The first line of the input contains a single integer T, indicating there are T test cases.

In each case, the first line contains two integers N and M, which denotes the number of nodes and queries.

Each of the following (N-1) lines contains a triple of integers (a,b,c), indicating there is an edge connecting node a and node b with a value of c.

The following M lines describe the queries. Each of the M lines consists of two integers A and B, indicating the starting node of Alice and Bob, respectively.

($2 \le N \le 100000, 1 \le M \le 100000, 1 \le a,b,A,B \le N, 1 \le c \le 1000$)

Output

For each query, output the maximum sum that Alice can obtain, one per line.

Sample Input

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

Sample Output

```
3
3
4
```

E. Deque

Today, the teacher gave Alice extra homework for the girl weren't attentive in his class. It's hard, and Alice is going to turn to you for help.
The teacher gave Alice a sequence of number(named A) and a deque. The sequence exactly contains N integers. A deque is such a queue, that one is able to push or pop the element at its front end or rear end. Alice was asked to take out the elements from the sequence in order(from A_1 to A_N), and decide to push it to the front or rear of the deque, or drop it directly. At any moment, Alice is allowed to pop the elements on the both ends of the deque. The only limit is, that the elements in the deque should be non-decreasing.
Alice's task is to find a way to push as many elements as possible into the deque. You, the greatest programmer, are required to reclaim the little girl from despair.

Input

The first line is an integer T(1≤T≤10) indicating the number of test cases.
For each case, the first line is the length of sequence N(1≤N≤100000).
The following line contains N integers $A_1,A_2,\ldots,A_N$.

Output

For each test case, output one integer indicating the maximum length of the deque.

Sample Input

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

Sample Output

```
7
5
3
```

F. Magic Ball Game

When the magic ball game turns up, Kimi immediately falls in it. The interesting game is made up of N balls, each with a weight of w[i]. These N balls form a rooted tree, with the 1st ball as the root. Any ball in the game has either 0 or 2 children ball. If a node has 2 children balls, we may define one as the left child and the other as the right child.
The rules are simple: when Kimi decides to drop a magic ball with a weight of X, the ball goes down through the tree from the root. When the magic ball arrives at a node in the tree, there's a possibility to be catched and stop rolling, or continue to roll down left or right. The game ends when the ball stops, and the final score of the game depends on the node at which it stops.
After a long-time playing, Kimi now find out the key of the game. When the magic ball arrives at node u weighting w[u], it follows the laws below:
1 If X=w[u] or node u has no children balls, the magic ball stops.
2 If X<w[u], there's a possibility of 1/2 for the magic ball to roll down either left or right.
3 If X>w[u], the magic ball will roll down to its left child in a possibility of 1/8, while the possibility of rolling down right is 7/8.
In order to choose the right magic ball and achieve the goal, Kimi wonders what's the possibility for a magic ball with a weight of X to go past node v. No matter how the magic ball rolls down, it counts if node v exists on the path that the magic ball goes along.
Manual calculating is fun, but programmers have their ways to reach the answer. Now given the tree in the game and all Kimi's queries, you're required to answer the possibility he wonders.

Input

The input contains several test cases. An integer T(T≤15) will exist in the first line of input, indicating the number of test cases.
Each test case begins with an integer N(1≤N≤10^5), indicating the number of nodes in the tree. The following line contains N integers w[i], indicating the weight of each node in the tree. (1 ≤ i ≤ N, 1 ≤ w[i] ≤ 10^9, N is odd)
The following line contains the number of relationships M. The next M lines, each with three integers u,a and b(1≤u,a,b≤N), denotes that node a and b are respectively the left child and right child of node u. You may assume the tree contains exactly N nodes and (N-1) edges.
The next line gives the number of queries Q(1≤Q≤10^5). The following Q lines, each with two integers v and X(1≤v≤N,1≤X≤10^9), describe all the queries.

Output

If the magic ball is impossible to arrive at node v, output a single 0. Otherwise, you may easily find that the answer will be in the format of $7^x/2^y$. You're only required to output the x and y for each query, separated by a blank. Each answer should be put down in one line.

Sample Input

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

Sample Output

```
0
0 0
1 3
```

G. Occupy Cities

The Star Wars is coming to an end as the Apple Planet is beaten by the Banana Planet. Captain Chen, the glorious leader of the Army of Banana Planet, has drawn up a plan to occupy all the cities on the Apple Planet. The Army of Banana Planet totally has P soldiers, and thus, Captain Chen can only conduct at most P soldiers to occupy the cities. The cities on the planet can be regarded as points on a 2D plane. What's more, there are some barriers on the planet, which can be seen as segments on the plane. When a soldier moves from city to city, he's not allowed to cross or touch the barriers. However, the soldiers are smart enough to go along the shortest paths between cities.

But these soldiers are just soldiers, whereupon they also need food to replenish their energy. A soldier needs one unit of food to move one unit of distance forward. Fortunately, all the cities have sufficient food supplies. When a soldier steps in a city, he will fill up his food bag. Invaders as they are, the soldiers will burn up all the food after filling his bag. And thus, each city can supply only one soldier.

When a soldier steps in a city, this city is occupied by the Army of Banana Planet immediately. Soldiers can also just pass by a city but not step in. In this case, this city is not occupied yet, and the food in the city would not be burned.

Captain Chen has an occupying schedule for his soldiers. If city A is arranged before city B on the schedule, city A must be occupied before city B. All the soldiers will strictly follow this schedule. During the occupying process, soldiers can be air-dropped to any positions on the plane as needed. After a soldier lands on the ground, he can only move on foot, and replenish his energy by the food in his bag. Note that their bags are full of food initially, and all bags have the same volume for soldiers.

You, the logistics minister of the army, are required to help the Captain to cut down the cost and determine the minimal volume of all P soldiers' food bags to finish occupying. All the requirements above should be fulfilled for sure.

Input

The first line contains an integer T(T≤50), indictaing the number of test cases.

Each test case begins with three integers n(0<n≤100), m(0≤m≤100) and p(0<p≤100), which respectively denotes the number of cities, barriers and soldiers.

The following n lines describe the cities' coordinates (x_i,y_i).

The next m lines, each with two pairs of integers (sx_i,sy_i) and (ex_i,ey_i), describe the two endpoints of each barrier.

The last line of each test case consists of n integers, describing the occupying schedule in order.

All the coordinates range from -10000 to 10000, and cities are labeled from 1 to n. You may assume that any two barriers will not have common points and cities will not be built on barriers.

Output

For each test case, output the minimal volume of soldiers' food bag, in accuracy of two decimal places. The answers should be printed one per line.

Sample Input

```
2

2 1 1
0 0
2 0
1 1 1 -1
2 1

4 2 2
0 1
5 1
8 0
1 -1
0 0 2 0
6 0 6 3
1 2 3 4
```

Sample Output

```
2.83
3.41
```

H. Park Visit

Claire and her little friend, ykwd, are travelling in Shevchenko's Park! The park is beautiful - but large, indeed. N feature spots in the park are connected by exactly (N-1) undirected paths, and Claire is too tired to visit all of them. After consideration, she decides to visit only K spots among them. She takes out a map of the park, and luckily, finds that there're entrances at each feature spot! Claire wants to choose an entrance, and find a way of visit to minimize the distance she has to walk. For convenience, we can assume the length of all paths are 1.
Claire is too tired. Can you help her?

Input

An integer $T(T \le 20)$ will exist in the first line of input, indicating the number of test cases.

Each test case begins with two integers N and $M(1 \le N,M \le 10^5)$, which respectively denotes the number of nodes and queries.
The following (N-1) lines, each with a pair of integers (u,v), describe the tree edges.
The following M lines, each with an integer $K(1 \le K \le N)$, describe the queries.
The nodes are labeled from 1 to N.

Output

For each query, output the minimum walking distance, one per line.

Sample Input

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

Sample Output

```
1
4
```

I. I-number

The I-number of x is defined to be an integer y, which satisfied the the conditions below:
1. y>x;
2. the sum of each digit of y(under base 10) is the multiple of 10;
3. among all integers that satisfy the two conditions above, y shouble be the minimum.
Given x, you're required to calculate the I-number of x.

Input

An integer T(T≤100) will exist in the first line of input, indicating the number of test cases.
The following T lines describe all the queries, each with a positive integer x. The length of x will not exceed 10^5.

Output

Output the I-number of x for each query.

Sample Input

```
1
202
```

Sample Output

```
208
```

J. 3-idiots

King OMeGa catched three men who had been streaking in the street. Looking as idiots though, the three men insisted that it was a kind of performance art, and begged the king to free them. Out of hatred to the real idiots, the king wanted to check if they were lying. The three men were sent to the king's forest, and each of them was asked to pick a branch one after another. If the three branches they bring back can form a triangle, their math ability would save them. Otherwise, they would be sent into jail.

However, the three men were exactly idiots, and what they would do is only to pick the branches randomly. Certainly, they couldn't pick the same branch - but the one with the same length as another is available. Given the lengths of all branches in the forest, determine the probability that they would be saved.

Input

An integer T($T \le 100$) will exist in the first line of input, indicating the number of test cases.

Each test case begins with the number of branches N($3 \le N \le 10^5$).

The following line contains N integers a_i ($1 \le a\_i \le 10^5$), which denotes the length of each branch, respectively.

Output

Output the probability that their branches can form a triangle, in accuracy of 7 decimal places.

Sample Input

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

Sample Output

```
0.5000000
1.0000000
```

K. Cards

Given some cards each assigned a number, you're required to select EXACTLY K cards among them.
While you select a card, I will check the number assigned to it and see if it satisfies some of the following conditions:
1. the number is a prime number;
2. the amount of its divisors is a prime number;
3. the sum of its divisors is a prime number;
4. the product of all its divisors is a perfect square number. A perfect square number is such a kind of number that it can be written as a square of an integer.
The score you get from this card is equal to the amount of conditions that its number satisfies. The total score you get from the selection of K cards is equal to the sum of scores of each card you select.
After you have selected K cards, I will check if there's any condition that has never been satisfied by any card you select. If there is, I will add some extra scores to you for each unsatisfied condition. To make the game more interesting, this score may be negative.
After this, you will get your final score. Your task is to figure out the score of each card and find some way to maximize your final score.
Note that 1 is not a prime number. In this problem, we consider a number to be a divisor of itself. For example, considering the number 16, it is not a prime. All its divisors are respectively 1, 2, 4, 8 and 16, and thus, it has 5 divisors with a sum of 31 and a product of 1024. Therefore, it satisfies the condition 2, 3 and 4, which deserves 3 points.

Input

The first line of the input contains the number of test cases T.
Each test case begins with two integers N and K, indicating there are N kinds of cards, and you're required to select K cards among them.
The next N lines describes all the cards. Each of the N lines consists of two integers A and B, which denote that the number written on this kind of card is A, and you can select at most B cards of this kind.
The last line contains 4 integers, where the ith integer indicates the extra score that will be added to the result if the ith condition is not satisfied. The ABSOLUTE value of these four integers will not exceed 40000.
You may assume $0<N\le10^3$, $0<K\le10^4$, $1\le A\le10^6$, $1\le B\le10^4$, $T\le40$ and the total N of all cases is no more than 20000. In each case there are always enough cards that you're able to select exact K cards among them.

Output

Output two lines for each test case.
The first line consists of N integers separated by blanks, where the ith integer is the score of the ith card.
The second line contains a single integer, the maximum final scores you can get.

Sample Input

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

Sample Output

```
1 3 2 2 2
11
```