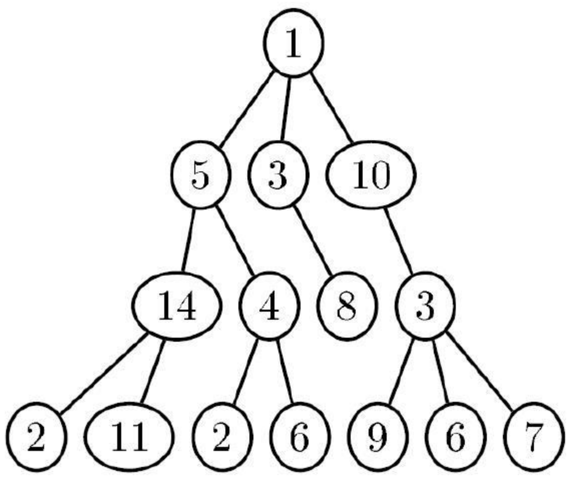
Problem 1. Throwing a Party

(Time Limit: 2 seconds)

Problem Description

Consider a company that has a hierarchical structure; that is, the "supervisor" relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is an integer. An example is as follows.



You are responsible to plan a party for the company. In order to make the party fun for all participants, you do not want an employee and his/her immediate supervisor to attend at the same time. For instance, the two people with conviviality ratings 11 and 14 in the above example are not supposed to show up together, albeit their ratings are the highest in the company. The goal is to maximize the sum of the conviviality ratings of the guests. For instance, in the above example, a party with the highest sum (i.e., 66) of conviviality has to exclude the five people with ratings 1, 3, 4, and 14.

1.There are n ($1 \leq n \leq 1000$) people in the company, each of them has a unique ID from 1 to n. The ID of the president is 1.

2.For each $i = 1,2,. . . ,n$, the conviviality rating r_i of the person with ID i is a positive integer no more than 1000.

Input Format

The first line of the input file contains an integer indicating the number of test cases to follow.

For each test case, the first line of input contains two integers *n* and r_1. For each *i* = 2,...,*n*, the *i*-th line of input contains two integers s_i and r_i, where s_i is the ID of the immediate supervisor of the person with ID *i*.

Output Format

For each test case, output the maximum sum of conviviality ratings of the guests.

Example

Sample Input:	**Sample Output:**

2
4 7
1 5
2 6
3 8
15 1
1 5
1 3
1 10
2 14
2 4
3 8
4 3
5 2
5 11
6 2
6 6
8 9
8 6
8 7
	15
	66

Problem 2. Regions

(Time Limit: 3 seconds)

Problem Description

Paul is the king of a country called Underwood. There are n cities in Underwood, which are connected by $n – 1$ roads. Each road is bidirectional, connects two different cities, and has a positive integer length. Furthermore, the $n – 1$ roads connect the n cities into a path.

There is a considerable population in each city of Underwood. Therefore, it is very difficult to manage every city by King Paul only. To improve the efficiency of management, Paul is planning to divide the cities of Underwood into several regions, each of which will be managed by a local administrator, according to the following conditions:

(1) The cities of each region A are *connected*. That is, the cities on the path connecting any two cities in A must also be contained in A.

(2) Each city belongs to exactly one region.

(3) The total population of any region is at least L.

To make a local administrator manage a region simpler, King Paul hopes to have as many regions as possible. Two regions are *adjacent* if and only if there is a road connecting two cities of them. King Paul also hopes that the distance between any two adjacent regions should not be too large. More specifically, King Paul wishes to maximize the number of regions; and in case there are several way to maximize the number of regions, he wants the way that minimizes the total distance between adjacent regions. Your task is to help King Paul to compute the maximum number of regions R along with the minimum total distance D between adjacent regions.

Consider the example shown in Figure 1, where *n* = 7 and *L* = 7. The nodes and edges represent the cities and the roads between the cities. A feasible way to divide the cities into regions is shown in Figure 2. The number of regions is three, where the populations are 7, 8, and 10 respectively. The total distance between adjacent regions is 1 + 6 = 7. Another feasible way is shown in Figure 3, which is the best way to divide the cities. The number of regions is still three, but the total distance between adjacent regions is only 1 + 3 = 4. Therefore, *R* = 3 and *D* = 4.

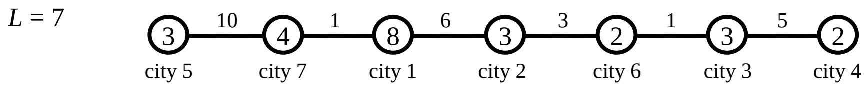


Figure 1.

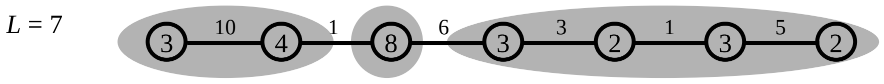


Figure 2.

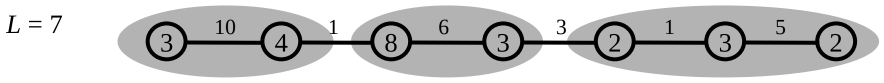


Figure 3.

Technical Specification

- The number of cities *n* is an integer between 1 and 500.
- The population of each city is at most 1000.
- The length of each road is at most 1000.
- The minimum population of a region *L* is at most 50000.

Input Format

The first line contains an integer $t \le 10$ indicating the number of test cases. The first line of each test case contains two integers *n* and *L*, the number of cities and the minimum population of a region. The cities are denoted by 1, 2, ..., *n*. Then, there is a line containinng *n* integers, where the *i*th integer is the number of population in city *i*. In the next *n* – 1 lines, each line contains three integers *i*, *j*, *d*, separated by a single space. It means that there is a road with length *d* connecting city *i* and city *j*, where $1 \le i, j \le n$ and $1 \le d \le 1000$.

Output Format

For each case, print the output in a single line. If there is no way to divide the cities into regions, print "-1". Otherwise, print two integers *R* and *D*. Use the format as in the Sample Output.

Example

Sample Input:	**Sample Output:**

2
7 7
8 3 3 2 3 2 4
7 5 10
1 2 6
3 6 1
2 6 3
3 4 5
1 7 1
2 13
3 2
1 2 8
	3 4
	-1

Problem 3. Job Selection

(Time Limit: 1 seconds)

Problem Description

The Advanced Computer Manufacturer (ACM) is a company that provides machine rental services for its clients to manufacture computer in a customized way. The core technology of the company is unique and there is only one set of equipment to be rented currently. Hence, while the company receives a lot of rental requests from its clients, it can only provide the service to one client at the same time. Consequently, it must reject some requests from the clients. In other words, two requests with overlapping durations cannot be both accepted. For each request, it is also known how much profit the company can make from the rental. Therefore, the objective of ACM is to select a set of requests that it can accept to maximize its total profit. Formally, let $\{(s_i,t_i,v_i)| \; 1\leq i\leq n\}$ be the set of requests, where s_i, t_i ($s_i< t_i$), v_i denote the starting time, the finishing time, and the profit of the ith rental request respectively. Then, a feasible solution is a subset $I\subseteq\{1,2,...,n\}$ such that for $i,j\in I$ and $i\neq j$, $t_i\leq s_j$ or $t_j\leq s_i$, and the objective is to find $\max\{\Sigma_{i\in I}\, v_i|\; I$ is a feasible solution$\}$.

Technical Specification

- The number of requests is from 0 to 10000
- For each request (s, t, v), we have $0\leq s<t\leq 100000$ and $1\leq v\leq 1000$

Input Format

The first line is an integer which indicates the number of test cases. Each test case consists of several lines. The first line of a test case contains a positive integer n that indicates the number of rental requests received by ACM. The second to the n+1st lines of a test case specify these requests, each of which contains 3 natural numbers s, t ($s<t$), v, which denote the starting time, the finishing time, and the profit of the rental respectively.

Output Format

For each test case, output the maximum total profit.

Example

Sample Input:	**Sample Output:**

2
5
14 69 50
7 80 68
44 46 29
10 47 7
26 31 53
4
0 3 25
8 9 40
1 3 27
6 8 34
	82
	101

Problem 4. Questionnaire

(Time Limit: 3 seconds)

Problem Description

Do you want to be a millionaire? Join our show Questionnaire! The Questionnaire is a kind of TV-show that participants need to answer question which matches the polling result from the audience. This season, the director decides to form all $m + n$ participants into two teams (first team has m participants and the second team has n participants), and pick out as many as possible pair among them into several head-to-head battle.

However, not all of the participants is suitable to have a battle to some others – that will make the show boring. The program director has chosen a set of pairs that could be formed in this show. Moreover, each participant has some certain personality so that the program director give each of them some exciting value v – the higher the value, the better the show.

In the first round, there are maximal possible pairs of participants chosen from the given set, any single participant can appear in this list at most once. The exciting value of the show is defined by the sum of all exciting values in paired participants.

You, as a concessionaire, in order to help the doctrinaire, you'll need to find out the maximum exciting value among all possible battle choices on the first round in this season.

Technical Specification

- $1 \leq m, n \leq 500$.
- Number of test cases $T \leq 50$
- All exciting values v satisfying $1 \leq v \leq 10^6$

Input Format

The first line of the input contains an integer T indicating the number of test cases. For each test case, the first line contains two integers m, n representing the number of participants in each team. The second line contains m integers – the exciting values of participants numbered $1, 2, \ldots, m$ in the first team. The third line contains n integers – the exciting values of participants numbered $1, 2, \ldots, n$ in the second team. There are m strings $s_1, s_2, \ldots, s_m$ follow. The i-th string $s_i = s_{i1}s_{i2} \ldots s_{in}$ is a sequence of zeroes and ones. $s_{ij} = 1$ iff i-th participant in the first team and j-th participant in the second team can be paired up.

Output Format

For each test case, output the maximum possible exciting value among all choosing maximum number of pair choices.

Example

Sample Input:	**Sample Output:**

2
3 3
1 2 3
4 5 6
001
100
010
3 3
1 2 3
4 5 6
001
111
001
	21
	16

Problem 5. Little Dora and Tableaux

(Time Limit: 3 seconds)

Problem Description

Little Dora is learning on integers and their ordering these days. She has a $m \times n$ sized rectangular grid board. In each cell of the grid there is a wooden cube marked by an integer from 1, 2, 3, …, to mn. Each time she scrambled the cubes, and place them back onto the board. Soon she found that the cubes are no longer being ordered!

However, Little Dora has discovered some good news: the numbers in each row and each column are (and will be) sorted, either in increasing or decreasing order. She call this property a “good ordered property” Now she has put the k smallest numbered cubes onto the board. Little Dora wondered, in how many ways that she can fill in all other cubes so that the board will be “good ordered”?

Technical Specification

- $1 \leq m, n \leq 10$
- The answer may be large, please output the solution modulo $10^9 + 7$.
- The number of test cases $T \leq 50$, most of them are having small dimensions in order to check your correctness.

Input Format

The first line contains an integer T indicating the number of the test cases. For each test case, there are two integers m, n in the first line. Then there are m rows each of them have n integers $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ separated by whitespaces. All integers are within the range $[0, mn]$. If the $a_{i,j} = 0$ that means the cell at i-th row and j-th column contains no cube now. It is guaranteed that first k positive integers are already in the board for some certain k (Possibly zero) and they appears at most one time in each given board.

Output Format

For each test case, output the number of possible ways to form the final “good ordered” board.

Example

Sample Input:	**Sample Output:**
5	2
3 3	12
1 2 3	0
4 5 0	64004
6 0 0	2
3 3	
0 0 0	
0 0 0	
3 2 1	
1 5	
1 2 0 0 3	
3 7	
1 2 3 4 0 0 0	
0 0 0 0 0 0 0	
0 0 0 0 0 0 5	
1 2	
0 0	