Bytecode 2012

Problems And Solutions

Problem Setters

Chakradarraju

Nithin Mahesh

Prasanna

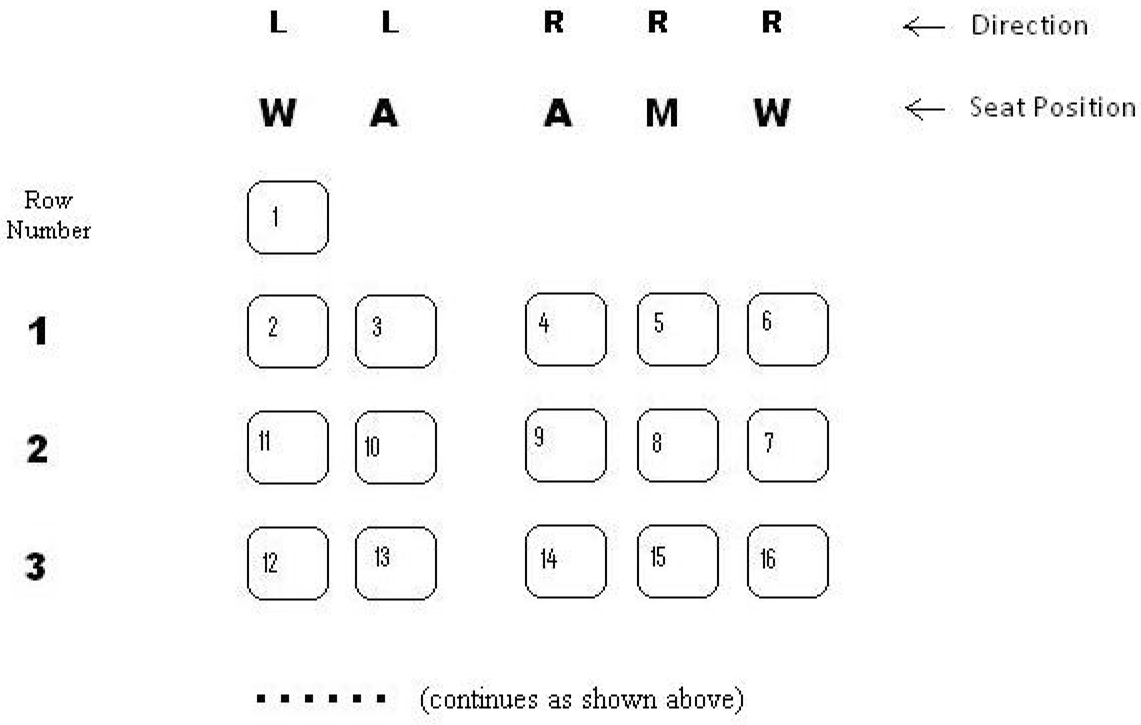
Problem Tester

SuhashVenkatesh

The Blind Passenger

The seats of an Ultra Deluxe Semi-sleeper bus in Thuvax country are numbered according to the following rules:

1. Seat number 1 is reserved for the bus conductor.
2. Remaining seats are numbered as shown in the figure below:



A blind person enters the bus with a reservation ticket mentioning his seat number. He meets you at the start of the passage and shows you his ticket. Based on the seat number, help him find his seat by telling him the row number (1,2,...), seat position(window or aisle or middle), and the direction(left or right).

Input

First line contains a positive integer t(t<=10^5) denoting the number of test cases.
Each of the next t lines contain a positive integer n(n<=10^9) denoting the blind person's seat number.

Output

Output one line for every test case in the following format:
If the seat number is same as the conductor's seat, print "poor conductor" else,
print <row_number><seat_position><direction>
row_number: represents the row number from the entrance (a positive integer)
seat_position: can be one of "W" or "A" or "M" for window, aisle and middle respectively.
direction: can be either "L" or "R" for left or right respectively.

Example

```
Input:
3
1
2
3

Output:
poor conductor
1 W L
1 A L
```

Solution

Since we know the seating arrangement of a single row, by using simple arithmetic operations we can answer the necessary details in constant order.

Complexity:O(1)

The Wild Wizard

In the country of Thuvax lived an old wizard named Chanithpra. He practiced a form of magic in which the power of spell came from the letters that were written on a piece of parchment.
When a sentence s is written, the intensity of the letters are as follows:
1. Intensity of 'a' is 1, 'b' is 2 and it continues till 'z' whose intensity is 26.
2. Intensity of 'A' is 51, 'B' is 52 and it continues till 'Z' whose intensity is 76.
3. Intensity of ' '(space) is 32.

The power of a spell written as a part of the sentence is equal to the sum of the intensities of the characters present in it modulo m.

Chanithpra finds an old parchment in Thuvax with a sentence s written on it.
Help him find the longest spell (i, i+1, ... j) in the sentence s whose power matches with k.
(0<=i<=j<length of s)

You need to print the starting index i(0<=i<=length-1) of such a spell in the sentence and the length of the spell.

If there are multiple such spells, find the minumum index i and if there is no such spell print -1.

Input

The first line contains a single positive integer t(1<=t<=100) denoting the number of test cases.
Each test case consists of 2 lines:
The first line consists of the sentence s(1<=length of s<=1000000).
The second line consists of two integers m and k (2<=m<=1000007 and 0<=k<m).

Output

For each test case, output one line containing 2 integers. The first is the minimum index i(0<=i<=s.length-1). The second is the length of the spell.
If no such spell is present, print -1

Example

```
Input:
1
Hello World
10 7

Output:
0 7
```

Solution:

Let A[i] be the sum of hash values between index 0 to i modulo m. Let the array hash[j] store the maximum index of occurence of j where j = A[i]+k. Now find the longest length of the string with earliest index by traversing backwards by using the hash array.

Complexity:O(length of the string)

The Dating Dress Problem

Gauthami has to get dressed for a date. She is going to meet Prasanna in the poshest restaurant in the country of Thuvax.
Prasanna arrives early and sees the outfit she has picked.
While waiting outside, Prasanna being a restless nerd calculates the number of ways she can get dressed.

The complete outfit she picks is represented as a string of digits. The digits represent the following details about the apparel:
(apparel = clothing or accessories or any other part of the outfit)

1 - a 2 piece apparel where first piece has to be worn before the other. It is followed by the description of the two pieces in order.

2 - a 2 piece apparel which can be worn in any order. It is followed by the description of the two pieces in order.

0 - a single piece of apparel which can be put on directly.

Help Prasanna calculate the number of ways in which Gautami can get dressed. Since Prasanna does not want to make Gauthami bored by telling her a huge number, he wants to tell her the number of ways modulo 1000000007.

For example, consider an outfit comprising of a shirt, a skirt into which the shirt has to be tucked in and a neck cloth worn over the shirt. It will be represented as 10200 and the number of ways she can get dressed is 2(She has to wear the shirt first after which she can wear either the skirt after the neck cloth or neck cloth after the skirt).

Input

First line is a positive integer T (T<=100) representing the number of testcases
It is followed by T lines, each containing a string S of digits (each digit will be one of 0,1,2) describing the outfit, |S|<=100,000

Output

Output for each test case should consist of one line representing the number of ways in which Gauthami can get dressed modulo 1000000007.

Example

```
Input:
3
10200
0
1102000

Output:
2
1
2

Explanation:
Let the input string be S.
1. Initially the outfit consists of a single apparel A. On 1(S[0]), A is split into A and B where B
can be worn only after A is worn. On 0(S[1]), A is worn as a whole. On 2(S[2]), B is split into
B and C where B and C can be worn in any order. Now on 0(S[3]), B is worn as a whole. On
final 0(S[4]), C is worn as a whole. So the final possibilities are A->B->C or A->C->B.
2. Initially the outfit consists of a single apparel A. On 0(S[0]), A is worn as a whole. Hence
the only possible way is A.
3. Initially the outfit consists of a single apparel A. On 1(S[0]), A is split into A and B where B
```

can be worn only after A is worn. Again on 1(S[1]), A is split into A and C where C can be worn only after A is worn. And on 0(S[2]), A is worn as a whole. On 2(S[3]), C is split into C and D which can be worn in any order. On 0(S[4]), C is worn as a whole. On 0(S[5]), D is worn as a whole. On 0(S[6]), B is worn as a whole. So now finally, there are two possibilities. A->D->C->B and A->C->D->B.

To make the problem more clear consider the graph 2100100, in which there are 4 nodes A and B are connected directly, C and D are connected directly. ACBD and CADB is also a valid ordering.

Solution

We can model this problem using graph. In this problem we are asked to find the number of topological sortings possible. Suppose there are two groups of nodes which are connected one after the other, the total number of ways of ordering the two groups together is the product of the number of ways of ordering the group of nodes separately since all the nodes in first group has to come before all the nodes in second group. Consider the case when you need to order two group of nodes in any order. Here order has to be preserved within each group but not between the groups. If group1 has n nodes and group2 has m nodes, then we have fill to them in n+m slots. So the number of ways of doing this ordering is (n+m)choose(n) since we have to choose any n slots to fill with group1 nodes and the remaining slots can be used for group2. By contracting the graph using the above two methods in the reverse order it expanded, we can find the total number of topological sorts of this graph.

Complexity: O(n)

Serve TheStreet

Shree is a very ambitious person and wants to start a new company "RAM COURIER SERVICE" in the country of Thuvax. He plans to open his first office in the Main Street of Thuvax to serve only the people of this street. The Main Street is an inclined road consisting of N buildings spaced out unevenly(building 1 at lowest level and building N at highest level). The distance between adjacent buildings is known to Shree.

The delivery costs incurred by him for various deliveries are calculated as follows:

To deliver a package of weight 'w' to a building downhill at a distance 'd' from the office, Shree spends w*d*1 units of money.
To deliver a package of weight 'w' to a building uphill at a distance 'd' from the office, Shree spends w*d*2 units of money.
To deliver a package of non-zero weight to his own building, Shree spends 10 units of money irrespective of the weight of the package.

Shree being an astute businessman wants to choose one of these buildings for his new office so that the total delivery cost incurred by him is minimum. Help Shree choose the building for his new office.

Input

The first line of the input consists of number of test cases t(1<=t<=20)

The first line of each test case consists of a single number N(1<=N<=10^6) which is the number of buildings in the main street.
The next line contains N integers representing the weight to be delivered at ith (1<=i<=N) building (0<=w[i]<=100).
The last line contains N-1 integers where the ith (1<=i<=N-1) integer represents the distance between i and i+1 th building(1<=d[i]<=100).

Output

Output a single line for each test case containing two integers separated by a space.
The first integer is the minimum delivery cost and the next integer is the building number where the office should be placed in order to incur that cost.
If more than one building has the same minimum cost, print the building labelled with the smallest number.

Example

```
Input:
1
5
1 2 3 4 5
1 1 1 1

Output:
30 4

Explanation:
If the office is placed in building 1, then the total cost is 10+2*1*2+3*2*2+4*3*2+5*4*2=90
If the office is placed in building 2, then the total cost is 1*1*1+10+3*1*2+4*2*2+5*3*2=63
If the office is placed in building 3, then the total cost is 1*2*1+2*1*1+10+4*1*2+5*2*2=42
If the office is placed in building 4, then the total cost is 1*3*1+2*2*1+3*1*1+10+5*1*2=30
If the office is placed in building 5, then the total cost is 1*4*1+2*3*1+3*2*1+4*1*1+10=30
```

Solution

For all buildings, compute the delivery costs for all packages to be delivered downhill and uphill separately. This can be computed using the recurrences below:
Let cw[i] represent the sum of all weights from w[0] to w[i];
Let dhcost[i] represent the cost of delivering packages downhill from buiding i.
Let uhcost[i] represent the cost of delivering packages uphill from buiding i.
dhcost[i] = dhcost[i-1] + cw[i-1]*d[i-1]
uhcost[i] = uhcost[i+1] + (cw[n-1]-cw[i])*d[i]*2
Now for all buildings, compute the total cost as follows:
If there is package of non-zero weight to be delivered at building i, cost[i] = uhcost[i] + dhcost[i] + 10

Else cost[i] = uhcost[i] + dhcost[i]
The minimum over all these costs gives the required minimum cost.

Complexity: O(n)

The Rail Network Renovation

The beautiful country of Thuvax has an ancient rail network connecting its 'n' cities using 'm' tracks. The King of Thuvax orders the Railway Minister, Poopsie to renovate the whole rail network. Due to Poopsie's past laziness, the tracks are in shambles and so he wants to rebuild(the tracks are functional again) some of the tracks and destruct(therefore, wont be available for usage) the rest completely.

The King wants to ensure that all the cities are connected after the whole network is renovated but being a miser, he wants to spend the least amount of money possible.
Help Poopsie find out the least amount of money using which the renovation project can be completed.

Input

The first line of the input contains the number of test cases 't'(1<=t<=100).

The first line of each test case consists of two numbers: number of cities 'n'(1<=n<=10^3) and number of tracks 'm'(0<=m<=10^6).
The next m lines contains description of each of the tracks and their corresponding costs of destruction and rebuilding.
Each of these m lines contains four integers 'a','b','d' and 'r'(1<=d,r<=1000000) representing a track between city 'a' and city 'b' with the cost of destruction and rebuilding being d and r respectively(cities are numbered from 1 to n).

Output

The ouput consists of t lines.

For each test case, print a single line consisting of a single integer that is the least amount of the money required to renovate the whole rail network.

Example

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

Output:
21
```

```
Explanation:
If we rebuild the tracks between 1 and 2, 2 and 3 and 2 and 4, and destroy the track between 1
and 4, the total cost of renovation is 6+8+5+2=21. This is the minimum cost of renovation.
```

Solution:

Since it is necessary that every road should be either renovated or destructed, we can renovate all tracks whose cost of renovation<=cost of destruction. For all other tracks, in the increasing order of their effective cost(Effective cost = cost of renovation - cost of destruction), we renovate it if it connects two cities that are not already connected.

Complexity: O(M log N)

The National game

Prasanna and Nithin had mastered Thuvax's national game, TicTacToe.

TicTacToe is a board game for two players, x and o, who take turns (alternatively) marking the spaces in a 3x3 grid with their signs 'x' and 'o' respectively. The x player goes first. The player who succeeds in placing three of his signs in a horizontal, vertical, or diagonal row wins the game and the game stops there.
While returning from a TicTacToe tournament, Prasanna and Nithin find a tictactoe like grid. They decide to continue playing from the state the board was in. Help Nithin find whether the next person to move will win/lose or whether the game will result in a tie (assuming both players play optimally), or if such a state is invalid.

Input

First line of the input contains a number T (1<=T<=10^6), followed by 4*T lines. The first line of each testcase is empty and the next 3 lines show the state of the game.

Output

Print win/lose/tie/invalid for each testcase on a separate line.

Example

```
Input:
3

oox
.x.
..x

xox
xoo
```

```
oxx

o..
...
...

Output:
lose

tie
invalid
```

Explanation:
Case 1: Player x can win by placing x in either last cell of the middle row, or the first cell of the last row. So the next player, o, cant prevent him from winning from this state.
Case 2: It is completed game, as you can see no one has won. So, it has ended in a tie.
Case 3: Player x should have started the game. Hence such a state can never be reached.

Solution:

Mark all states as invalid. Do the following recursive procedure starting with an empty board with Player x to move:

```
Procedure Tic Tac Toe

Begin

If this state is already processed, return its result.

Make this state valid.

If someone has won already, next player is going to lose, so it is a losing state and the game wont proceed further.

For every empty cell in the board, place the sign of the player to move in that cell and call the recursive procedure of that state with other player to move.

If at least one of the next states is a losing state, then current state is a winning state.

If all next states are winning states, then current state is a losing state.

If both of the condition doesnt hold then the state will become tie.

Store the result for this state and return it.

End
```

Now for every query, print its state.

Divide and conquer

The King of Thuvax decided to conquer the neighbouring country of Silicon Valley. In order to conquer a country, he has to successfully invade every city in the country. But he always doesn't get into action unless he plans it properly. He knows that severing a city's communications with all other cities before invading it is the only way to conquer it. He has the information regarding how the cities communicate with each other through his spy. The spy has lived in Silicon Valley right from the time when it was a single city, much before its development into a country. He gives information about how every new city established communication links with other cities. The spy's information is represented as below.

Every city can establish its communications with the help of only one other city. It can do it in one of the three ways.

Type 1-It can establish a single communication link with another city 'c'.

Type 2-It can establish communication links with all cities that currently communicate with another city 'c', but it can't have a communication link with 'c'.

Type 3-It can establish communication links with all cities that currently communicate with another city 'c' and with 'c' also.

All communication links are two-way. Having this information, he finds out that it is not possible for him to invade Silicon Valley in a fair way. So he hires a terrorist gang to bomb some cities and destroy them so that he can invade the country successfully. But the terrorists demanded a huge amount of money for bombing a city. Help the King to find out the minimum number of cities that should be bombed so that he can conquer the country of Silicon Valley successfully.

Input

The first line consists of number of test cases t(1<=t<=10)

The first line of each test case consists of the number of cities n(1<=n<=10^4) in Silicon Valley.

Each of the next n-1 lines consists of two integers x and c.

The ith(1<=i<=n-1) line represents that city i+1 establishes a type 'x' communication link with the help of the city 'c'(1<=c<=i).(The country initially contains city 1 alone.)

Output

Output one line for each test case containing the minimum number of cities that should be bombed so that the King can conquer the country

Example

```
Input:
2
2
1 1
3
1 1
3 1

Output:
1
2
```

Explanation:
1. Here city 2 has communication links with city 1. So to conquer the country, The King has to bomb one of these cities and then invade the remaining city.
2. Here city 2 has communication links with city 1 and city 3 establishes communication links with city 1 and city 2. So The King has to bomb any of these two cities and then invade the third city to conquer the country.

Solution:

It can be easily deduced that we need to find the Vertex Cover of the given graph. The graph formed using the given three operations is a Distance-Hereditary Graph. Vertex cover of a distance-hereditary graph can be found in polynomial time.

Generate an Extension Tree for the graph from the input as follows:

Let the tree initially consist of node 1 alone.

If a new node 'i' establishes communications of type 'x' with another node 'c', make 'i' as the rightmost child of 'c' and label the edge between 'c' and 'i' as 'x'.

Let us generalize the Vertex Cover Problem as given a Vertex Set S and a subset of it R, find a subset S of V, such that G(V-S) contains no edges and R is a subset of S. When R={}, it is the general Vertex Cover Problem.

Let G(i) represent the graph formed by the Extension Tree rooted at node 'i'.

Let G(i,h) represent the graph formed by the Extension Tree rooted at node 'i' excluding the children to the left of 'h' and R={}.

Let GR(i,h) = G(i,h) - G(h)

Let TS(i) represent all nodes below node 'i' including 'i', that can be reached from node 'i' using edges of type 2 and type 3 alone.Note that TS(i) always contains i.

Let VC(i) represent the vertex cover of the graph G(i) with R={}.

Let VTS(i) represent the vertex cover of the graph G(i) with R=TS(i).

We can find out the vertex cover of the original graph from its Extension Tree as follows.

For a node 'l' with no children, it can be easily deduced that VC(l)={} and VTS(l)={l}.

VC(i) = VC(i,k) where k is the leftmost child of k.

Now the vertex cover for G(i,h) can be found from graphs G(h) and GR(i,h):

For G = G(i,h), G1 = G(h) and G2 = GR(i,h)

If the edge between i and h is of type 1, VC(G) = min{VC(G2) U VTS(G1), VC(G1) U VTS(G2)}, VTS(G) = VTS(G1) U VC(G2).

If the edge between i and h is of type 2, VC(G) = VC(G1) U VC(G2) and VTS(G) = VTS(G1) U VTS(G2).

If the edge between i and h is of type 3, VC(G) = min{VC(G2) U VTS(G1), VC(G1) U VTS(G2)}, VTS(G) = VTS(G1) U VTS(G2).

Thus, for the whole graph, it can be calculated in the reverse order in which the input was given.

For proof of the procedure refer: http://www.springerlink.com/content/e445670215750333/

Mirror Number

A number is called a Mirror number if on lateral inversion, it gives the same number i.e it looks the same in a mirror. For example 101 is a mirror number while 100 is not.
Given two numbers a and b, find the number of mirror numbers in between them (inclusive of a and b).

Input

First line contains T, number of testcases<= 10^5.
Each testcase is described in a single line containing two numbers a and b.

0 <= a<=b <= 10^44

Output

For each test case print the number of mirror numbers between a and b in a single line.

Example

```
Input:
3
0 10
10 20
1 4

Output:
3
1
1

Explanation:
1. The mirror numbers between 0 and 10 are 0,1,8.
```

2. The mirror number between 10 and 20 is 11.
3. The mirror numbers between 1 and 4 is 1.

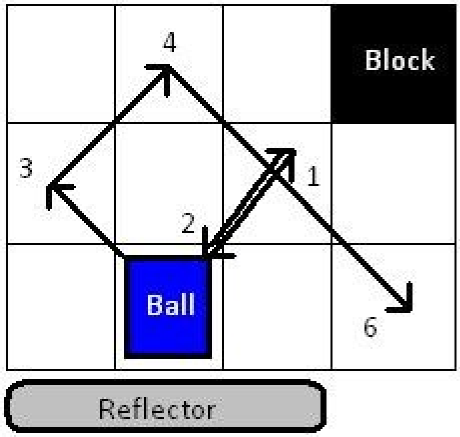
Solution

It is a classical f(b)-f(a) type of problem. If you can find the number of mirror numbers below a number x then you can find the number of mirror numbers in any range (a,b). We can use Dynamic Programming technique to precalculate the number of mirror numbers of each length. Since the mirror number has to be a palindrome, looking at the digits in one half of the number and the precalculated values we can calculate the total number of mirror number numbers less than x.

Complexity: O(number of digits)

The Lazy Gamer

In the kingdom of Thuvax, a mobile game called Thuball became very popular.
The description of the game is as follows:
A mobile phone made in Thuvax has a screen made of X*Y pixels. Thuball is played by throwing a ball (of size 1 pixel) at an angle of 45 degrees from the bottom of the screen. The ball changes its direction if and only if it hits the left/right/top edge of the screen or an obstacle on its way.
There are N fixed obstacles (of size 1 pixel each) on the screen. There are no obstacles in the bottom-most row of the screen.
If the moving ball's edge comes in contact with an obstacle's edge, it gets reflected by 90 degrees.(Priority High)
If the moving ball's corner comes in contact with an obstacle's corner, it is reflected by 180 degrees. (Priority Low)
In both reflections the obstacle doesn't move.
There is a reflector of length L pixels placed below the bottom row, which reflects (akin to mirror reflection, ie 90 degrees) the ball incident on it.The ball can start from any cell on the bottom row provided that the reflector is below it.
The game finishes only when the ball reaches the bottom row and the reflector is not present below it. The distance travelled by the ball for moving to a diagonal cell is assumed to be 1. The total distance (in pixels) travelled by the ball is the score attained during a round until the game finishes.
If the game does not finish, the score is 0.



Gosu, a Thuvaxian was very fond this game. He had set the highest score in Thuball in his friend Visu's phone. Not happy with the situation where he didn't have the highest score, Visu decided to recruit you to do his dirty work.
Visu is lazy and doesn't want to move the reflector after starting the game. But, he wants to ensure that he gets the highest score possible with a reflector of length L when its kept stationary in any position.
Help Visu out by telling him the maximum score he can get without moving the reflector for Q different lengths.

Input

First line contains two integers X and Y, which are the height and width of the screen. (1<=X,Y<=1000) Second line contains the number of obstacles N followed by N lines with two integers xi and yi, position of each obstacle. Consider the bottom-left corner of the screen as origin and the top-right corner to have the coordinates:(X-1,Y-1). The next line contains Q, for the number of cases (for different lengths of the reflector).(1<=Q<=1000) Followed by Q lines with one positive integer L, representing the length of the reflector.

Output

For each query output the maximum score Visu can achieve without moving the reflector in a single line.

Example

Input:

3 4

1

2 3

1

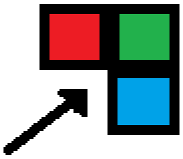
3

Output:
6

Explanation:
The reflector is kept between 0 and 2 and the ball is started from (0,1) in north-east direction(towards top-right). The ball traces the path as shown in the figure. One can see that the distance travelled is 6. This is the maximum score one can attain with a stationary reflector of size 3.

Solution

Solution for this problem can be written in two phases: in first phase calculate for each point and direction, which bottom point, direction it will reach, and time it will take. Then in second phase, we find the maximum score that can be achieved for every length of the reflector.
The first phase can be solved by simulating the entire process. For this we start from each column on the base and check where it exactly ends. The trickier part of the problem is the direction change when there is a collision with blocked cells. This can be done in the following way.



If R and B cells are blocked the ball reflects back 180 degree. The same applies when only green is blocked. If R is blocked then there will be 90 degree reflection and if the B is blocked then it will be 270 degree reflection. If there is no blocked cell then it continues as it is. By using this property we can calculate where the ball ends when it starts from a point in the base.

In the second phase, let us initialise the maximum score for all lengths to be 0. Starting from every point and direction, we remember leftextreme and rightextreme the ball has reached. For each hop, if it reaches new extreme, suppose leftextreme, then we can have the paddle from rightend of the screen to one point before new leftextreme and still hold the ball for all previous hops and let it down now. So update all lengths between previous maxlength reached to the current length-1, with the score to reach this point, similarly for rightextreme calculate current length with leftend and rightextreme. For updation of a range of lengths with current score, data structures like segment tree can be used.
After doing the operation for all points and direction, the data structure will have the answer for each length.
Complexity for phase 1: O(X*Y)
Complexity for phase 2: O(Y*Y*log Y)
Complexity for each query from segment tree: O(log Y)

The Great Escape

A map consists of N checkpoints (numbered from 1 to N) connected by M one way roads. A thief is at checkpoint S. He wants to move to checkpoint D. The police, guessing that the thief will move through the route that takes him the least time to reach D from S, have called for security alerts to be placed in all the roads of all such routes. The thief wants to reach D without passing through any of those security alerted roads in the least possible time. If there are multiple such routes, he wants to travel so as to cross a minimum number of checkpoints. Find the minimum time required by the thief to reach D from S, the minimum number of checkpoints in such a route(inclusive of S and D) and the number of such routes available. Since the number of such routes may be huge, print the number of such routes modulo 1000000007. If there is no path from S to D print -1.

Input

The first line of the input consists of a single integer t representing the number of test cases(1<=t<=300)
The first line of each test case consists of two integers N, M where N is the number of checkpoints and M is the number of roads. (1<=N <= 500 and 1<=M <= 10^4)
The next M lines consist of three integers x,y,t where x and y represent that the road can be used to travel to checkpoint y from checkpoint x in time t (t <= 100) (1<=x,y<=N)

The last line contains S and D (source and destination which are not the same checkpoints)

For any pair of checkpoints (x,y), there will not be more than one road from x to y.

Output

For each test Case, output a single line containing 3 integers x,y,z. Where x is the least amount of time needed to travel from S to D without using any of the security alerted roads, y is the minimum number of checkpoints in such a route and z is the number of such routes modulo 1000000007.

Example

```
Input:
1

3 3

1 3 2

1 2 2

3 2 4

1 2

Output:
```

```
6 3 1

Explanation:
The fastest route from 1 to 2 is 2. So avoiding that route, the fastest route is 1->3->2 which
takes a time of 6. This route contains 3 checkpoints and there is only one such possible route
with time taken as 6 and number of checkpoints as 3.
```

Solution:

First find the shortest path in the graph from S to D and by some means remember all such paths. Then erase all the edges in all those paths from the graph. Now again use dijkstra to calculate the path and the number of ways. Number of ways to reach a particular node in minimum time with a minimum number of nodes can be calculated by storing the same for the previous states .

Complexity O(M+NlogN)

The Nerd Factor

Prof.Venky handles Advanced Topics in Algorithms course for a class of 'n' students. He is always known for his unsolvable question papers. Knowing that it is impossible to pass his subject in a fair manner, one of the students of his class, Vishy, finds out from his seniors that Prof.Venky won't be able to find out if at least 'k' students together discuss and write the answers and thereby all of them can pass. Hence they decide to divide the whole class into a number of groups so that everyone passes. But all the students are fighting over forming the groups. So Puppala, one of the nerdy students in the class, decides that he will compute all possible ways that they can form the groups and number them, and finally choose one of those numbers at random and go ahead with that way. Now it is your duty to help Puppala find the number of ways that they can form such groups.

Puppala is incapable of reading big numbers, so please tell him the answer modulo 1000000007

Input

The first line contains the number of test case t(1<=t<=10^6).
Followed by t lines for each case.
Each test case contains two integers 'n' and 'k' separated by a space(1<=k,n<=1000)

Output

For each test case, print a single line containing one positive integer representing the number of ways modulo 1000000007.

Example

```
Input:
3
2 1
4 2
6 2
```

```
Output:
2
2
4

Explanation:
1. Different ways are {1,1} and {2}
2. Different ways are {2,2} and {4}
3. Different ways are {2,2,2}, {3,3}, {2,4} and {6}
```

Solution:

The given question can be rephrased as given two numbers n and k, find the number of ways to express n as sum of certain numbers where each number is atleast k.
This function can be defined as

```
if(k>n) f(n,k) = 0
if(k==n) f(n,k) = 1
otherwise f(n,k) = f(n,k+1)+f(n-k,k)(smallest number in the partitions is k + smallest number in the partition is greater than k)
```

This can be done with precomputing the values of all n and k according to the constraints. Each query can be answered in constant order

Complexity for precomputing:O(n*m)
Complexity for each query:O(1)