Southeastern European Regional Programming Contest
Bucharest, Romania – Vinnytsya, Ukraine
October 20, 2018

Problem J

Rabbit vs Turtle

Input File: standard input
Output File: standard output
Time Limit: 0.4 seconds (C/C++)
Memory Limit: 256 megabytes

A rabbit and a turtle decided to race each other. Since the turtle is from Craiova and the rabbit is from Ardeal, the turtle is obviously faster than the rabbit. Our goal is to help the rabbit win the race.

The race is held on a graph with N nodes and M vertices. The race starts at node 1 and ends at node N. Both the rabbit and the turtle decided beforehand to select the path which they are going to use (each one of them with his own path). Therefore, they know the graph, the paths and also the time it takes for each one of them to cross each edge of the graph.

The turtle may be faster than the rabbit, but it's still a turtle (let's call it George). George selects some nodes on his path where he decides to sleep for a while. If at any moment in time he realizes that the rabbit cheats, George will stop sleeping and will go to the finish without any further rest.

The rabbit (let's call it Stan) has only one advantage. Stan has a great-great-great-. . . ...-great grandmother who is a fox, therefore he has a sly side. Stan does not intend to keep his promise about the path (but George does). At some point, his plan is to change the path he is initially assigned and take the shortest path to node N. The only problem is that he has to be smart about it. The moment George discovers that Stan is cheating, he will stop his sleeping activities, which is not good.

Stan can only change his path when he is in a node (obviously not when he is on an edge). He does not know George's sleeping schedule, but you do!!! Compute the number of moments when Stan can change his path such that he will win the race. The moment Stan cheats, George will discover immediately as long as he is not sleeping. If he is sleeping during that time, he will find out upon waking up.

Input

The first line contains two numbers N and M. The next M lines will describe the graph by (A, B, T, R): there is an edge from node A to node B. The turtle will cross the edge in T units of time, while the rabbit will do it in R units of time. The i-th such edge is considered to be the edge with index i.

The next line contains a number P_T, the number of nodes in George's path. The next P_T lines will describe the path by (edge_index, sleep): the index of the next edge George is going to use and the number of time units he is going to sleep after using that edge. The sleeping value for the last edge is irrelevant since the turtle will already reach the finish.

The next line contains a number P_R, the number of edges in Stan's initial path. The last line will contain P_R values, the indices of edges Stan is going to use.

Output

The output contains on the first line one value: x, the number of moments (nodes on the path) when Stan can cheat. The second line contains x numbers in **ascending order**, the indices of the nodes where Stan can cheat.

Constraints

- $2 \leq N \leq 100.000$
- $1 \leq P_R, P_T < 100.000$
- $1 \leq M \leq 200.000$
- $1 \leq T, R \leq 1.000.000.000$

- $0 \leq sleep \leq 1.000.000.000$
- The edges from the input that describe the paths are given in the correct order.
- The nodes in the turtle's path do not repeat.
- The nodes in the rabbit's path do not repeat.
- If the rabbit and the turtle arrive at the finish in the same time, the rabbit wins.
- If the rabbit cheats in the exact moment when the turtle goes to sleep, the turtle will first fall asleep and only after waking up he will realize about the cheating.
- The rabbit is considered to cheat ONLY if he changes the direction of the path and the new path that he will take is strictly faster than the original one (otherwise, there is no point to cheat)
- The cheating path may have common nodes with the original path. The only restriction is that in the exact moment of cheating, the rabbit has to take a different node. He may even go back to the previous node if he has an edge that allows that.

Sample input	Sample output
8 12 1 2 2 10 2 3 1 10 3 8 2 10 1 4 10 3 4 5 10 2 5 6 10 4 6 8 10 2 1 7 10 5 4 7 10 2 5 7 10 2 6 7 10 1 7 8 10 1 3 1 3 2 2 3 0 4 4 5 6 7	2 4 5
6 6 1 4 1 3 4 6 1 1 4 2 1 6 2 6 6 6 3 4 2 3 1 3 4 5 2 1 2 2 0 4 6 5 3 4	0