Task 1: Competition (`competition`)

You are Nilan, teaching a class of n students. Recently, each student sat for two examinations — Physics and Biology. The scores of the i-th student in Physics and Biology are denoted by $A[i]$ and $B[i]$ respectively. Each score is a non-negative integer not exceeding 10000, i.e. $0 \leq A[i], B[i] \leq 10000, \forall i \in [1, n]$.

Based on their results, you are to select a and b students to participate in the Physics and Biology categories of a Science competition respectively. To give all students a chance to represent their school, each student must be enrolled in exactly one of the categories.

Naturally, the school wants to maximise its chances of winning by sending the strongest team possible. This is achieved when the sum of Physics scores of the a Physics students combined with the sum of Biology scores of the b Biology students is maximised.

As their teacher, determine the maximum possible combined score of the Physics and Biology teams!

Input

Your program must read from standard input.

The first line of the input contains 3 integers n, a and b denoting the total number of students, the size of the Physics team and the size of the Biology team. It is guaranteed that $a + b = n$ i.e. all students take part as either a Physics or a Biology competitor.

2 lines will follow. The first line contains n integers, i.e $A[1], A[2], \ldots, A[n]$. The second line contains n integers, i.e $B[1], B[2], \ldots, B[n]$.

Output

Your program must print to standard output.

The output should contain a single integer on a single line, the maximum combined sum of all the Physics scores of the Physics team, and the Biology scores of the Biology team.

Subtasks

The maximum execution time on each instance is 1.0s, and the maximum memory usage on each instance is 256MiB. For all testcases, the input will satisfy the following bounds:

- $1 \leq n \leq 10^5$
- $0 \leq a, b \leq n$
- $a + b = n$
- $0 \leq A[i], B[i] \leq 10000, \forall i \in [1, n]$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	29	$1 \leq n \leq 20$
2	22	$B[i] = 0, \forall i \in [1, n]$
3	49	-

Sample Testcase 1

This testcase is valid for subtasks 1 and 3.

Input	Output

`3 1 2`
`5 3 4`
`7 1 4`
	`14`

Sample Testcase 1 Explanation

The optimal way of choosing the school team is to assign the second student to the Physics team and the first and third students to the Biology team. The combined Physics score of the Physics team is 3, while the combined Biology score of the Biology team is $7 + 4 = 11$. The total score is thus $3 + 11 = 14$. It is impossible to achieve a total score higher than 14, thus 14 is our answer.

Sample Testcase 2

This testcase is valid for all subtasks.

Input	Output

`5 3 2`
`5 6 6 5 1`
`0 0 0 0 0`
	`17`

Sample Testcase 2 Explanation

The optimal way of choosing the school team is to assign the first three students to the Physics team and the last two students to the Biology team. The combined Physics score of the Physics team is $5 + 6 + 6 = 17$, while the combined Biology score of the Biology team is $0 + 0 = 0$. The total score is thus $17 + 0 = 17$. It is impossible to achieve a total score higher than 17, thus 17 is our answer.

Task 2: Departure (`departure`)

The final match of the prestigious national football championship has finally ended. It is now midnight, and football fans from all over the country were gathered in the national stadium, the largest sports venue in the whole of Mouseland, and are now leaving the venue to head back home by bus.

Mouseland may be modelled as a very long straight road in the east-west direction, and its inhabitants live in houses along this road. As the national stadium is the most important building in Mouseland, it is located at kilometre zero of this road, and all locations are specified relative to the stadium as a single number. Specifically, locations to the east of the stadium use positive numbers, and locations to the west of the stadium use negative numbers. For example "10" refers to the location ten kilometres east of the national stadium, while "-4" refers to the location four kilometres west of the national stadium. The stadium is at location "0".

There are N public bus routes along this road. Each bus route runs a single bus each day. The i^{th} bus travels from location S_i to E_i at a constant speed and direction, starting at midnight and reaching its destination at midnight the next day. Buses leave at midnight every day, so for every bus route, there is an identical bus that leaves its start location every day, travelling to the same destination. People can board buses, alight from them, and transfer between buses (at the same location) instantly and at any point along the bus route — they can hop on and off while the bus is moving, anywhere along the road. Note that the speed of a bus depends on the distance it needs to travel, and that bus routes are unidirectional (i.e. there might not be another bus that travels in the opposite direction of a given bus).

There are M people that left the national stadium at midnight. The j^{th} person lives at location P_j. For each of the M people, what is the minimum amount of time they need to get home by bus?

Input

Your program must read from standard input.

The first line contains two integers, N and M, which represent the number of buses and number of people respectively.

The next N lines each contain two integers. The i^{th} of these N lines contains integers S_i and E_i, which represent the start location and destination location of bus route i^{th} respectively.

The final line contains M integers. The j^{th} integer represents the location of the j^{th} person's home.

Output

Your program must print to standard output.

The output should contain exactly M lines. The j^{th} line should contain two integers, A_j and B_j, such that $\frac{A_j}{B_j}$ is a fraction in its simplest form (i.e. $\gcd(A_j, B_j) = 1$) representing the minimum number of days the j^{th} person needs to get home by bus. It is guaranteed that every person is able to eventually get home by bus.

Note that you need to report the exact number of days required, including the fractional part if any. For example, if a person can get home at the earliest at noon on the fourth day (so it took exactly 3.5 days to get home), then the required fraction is $\frac{7}{2}$, so you should output "7 2".

Implementation Note

As the input lengths for subtasks 4, 5, 6, and 7 may be very large, you are recommended to use C++ with fast input routines to solve this problem. The scientific committee does not have a solution written in Java or Python that can fully solve this problem.

C++ and Java source files containing fast input/output templates have been provided in the attachment. You are strongly recommended to use these templates.

If you are implementing your solution in Java, please name your file `Departure.java` and place your `main` function inside `class Departure`.

Subtasks

The maximum execution time on each instance is 4.0s, and the maximum memory usage on each instance is 256MiB. For all testcases, the input will satisfy the following bounds:

- $1 \leq N \leq 10^6$
- $1 \leq M \leq 10^6$
- $-10^6 \leq S_i, E_i \leq 10^6$ for all $1 \leq i \leq N$
- $-10^6 \leq P_j \leq 10^6$ for all $1 \leq j \leq M$
- $S_i \neq E_i$ for all $1 \leq i \leq N$
- $P_j \neq 0$ for all $1 \leq j \leq M$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	10	$N \leq 10^4$, $M \leq 10^3$, $\text{sign}(S_i) \neq \text{sign}(E_i)$ for all $1 \leq i \leq N$
2	14	$N \leq 10^2$, $M \leq 10^3$
3	12	$N \leq 10^3, M \leq 10^5$, $\frac{A_j}{B_j} \leq 10^3$ for all $1 \leq j \leq M$, for any x, $\min\{S_i, E_i\} \leq x \leq \max\{S_i, E_i\}$ for at most 10^2 choices of i
4	8	$M \leq 10^3$, for any x, $\min\{S_i, E_i\} \leq x \leq \max\{S_i, E_i\}$ for at most 10^4 choices of i
5	15	$M \leq 10^4$, $\frac{A_j}{B_j} \leq 10^2$ for all $1 \leq j \leq M$
6	13	$\text{sign}(S_i) \neq \text{sign}(E_i)$ for all $1 \leq i \leq N$
7	28	-

Note:

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

In other words, for any integer x, $\text{sign}(x)$ is 1 if x is positive, 0 if x is zero, and -1 if x is negative.

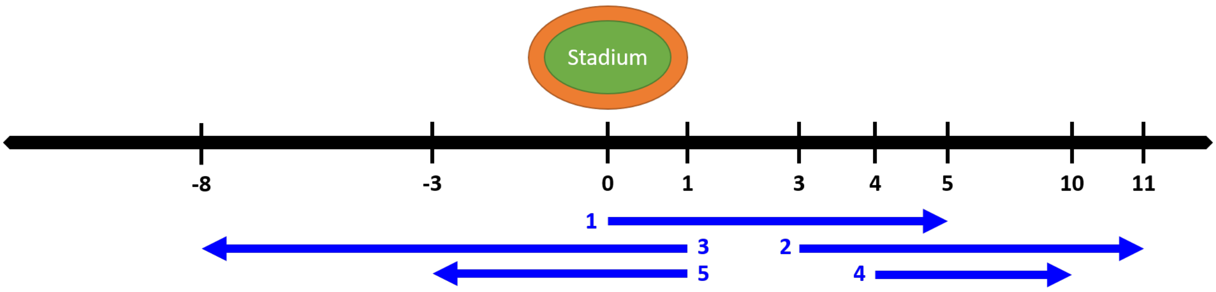
Sample Testcase 1

This testcase is valid for subtasks 2, 3, 4, 5, and 7.

Input	Output

`5 8`
`0 5`
`3 11`
`1 -8`
`4 10`
`1 -3`
`1 2 5 6 8 -3 -7 11`
	`1 5`
	`2 5`
	`1 1`
	`4 3`
	`13 8`
	`4 9`
	`8 9`
	`2 1`

Sample Testcase 1 Explanation



The above diagram (not to scale) shows the location of the stadium on the road, as well as all the bus routes (represented by blue arrows, and numbered for convenience).

To get to location 1, person 1 can take bus 1 from the stadium directly to location 1, and this takes $\frac{1}{5}$ of a day. There is no way to reach location 1 faster than this, so the output for person 1 is `1 5`.

To get to location 8, person 5 can take bus 1 from the stadium to location 3, and this takes $\frac{3}{5}$ of a day. They stay overnight at location 3, and take bus 2 from location 3 to location 8 the next day, thus reaching at $\frac{5}{8}$ of the second day. Since we need to measure the time taken from the start of the first day, the total number of days is $1 + \frac{5}{8} = \frac{13}{8}$. There is no way to reach location 8 faster than this, so the output for person 5 is `13 8`. Note that person 5 could instead stay overnight anywhere between location 3 and location 5, including at non-integer locations.

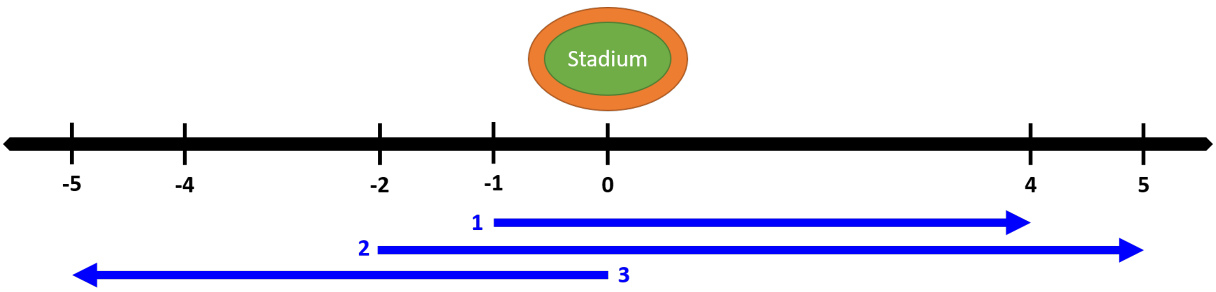
To get to location -7, person 7 can take bus 3 from the stadium directly to location -7, and reaches at $\frac{8}{9}$ of the first day. Note that even though they spent some time waiting at the stadium for the bus, that waiting time should be counted too. There is no way to reach location -7 faster than this, so the output for person 7 is `8 9`.

Sample Testcase 2

This testcase is valid for all subtasks.

Input	Output
3 2 -1 4 -2 5 0 -5 4 -4	6 7 4 5

Sample Testcase 2 Explanation



The above diagram (not to scale) shows the location of the stadium on the road, as well as all the bus routes (represented by blue arrows, and numbered for convenience).

To get to location 4, person 1 can take bus 1 to location 1.5, and instantaneously transfer to bus 2 to reach their destination at $\frac{6}{7}$ of a day. There is no way to reach location 4 faster than this, so the output for person 1 is `6 7`.

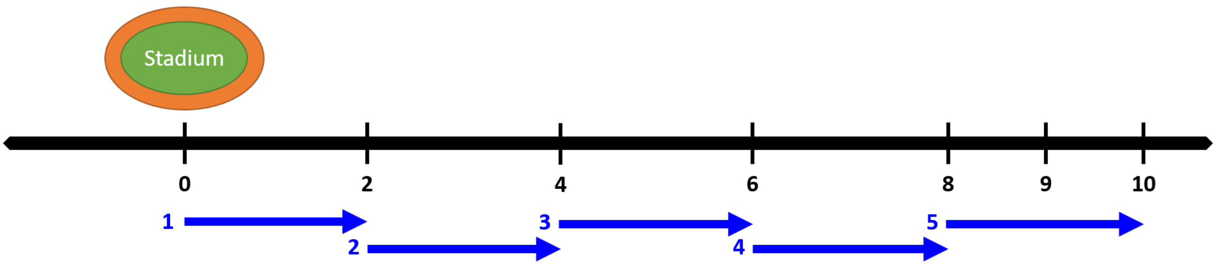
To get to location -4, person 2 can take bus 3 directly to location -4, and this takes $\frac{4}{5}$ of a day. There is no way to reach location -4 faster than this, so the output for person 2 is `4 5`.

Sample Testcase 3

This testcase is valid for subtasks 2, 3, 4, 5, and 7.

Input	Output
5 3 0 2 2 4 4 6 6 8 8 10 9 10 10	9 2 5 1 5 1

Sample Testcase 3 Explanation



The above diagram (not to scale) shows the location of the stadium on the road, as well as all the bus routes (represented by blue arrows, and numbered for convenience).

All people need to take the five buses in sequence to get to their destinations.

Task 3: Truck (`truck`)

In a faraway world, there are N towns, in these towns are heroes and they are preparing for an impending battle. However, while preparing their stores, they needed more gold! They needed money to buy more food and weapons. Thus, to deliver gold between towns, the heroes utilise their safest and most reliable transportation vehicles, Trucks.

The N towns are connected via $N - 1$ roads in a way such that there is **exactly one path between any two towns using one or more roads**. These roads are numbered from 1 to $N - 1$ each having their own length D_i. Also, there is a gatekeeper on each road which will collect a certain amount of gold bars as a toll for using the road, the toll for each road can be different and vehicles are required to pay this toll before using the road. **Notably, the i^{th} road connects towns A_i and B_i, has distance of D_i and toll cost of T_i.**

Obviously, the heroes have to pay for the fuel costs of travelling in the trucks. The amount of fuel used by a truck depends on the amount of gold it is currently carrying. **Notably, if the truck is carrying X gold bars and travels 1 unit distance, it will use up X units of fuel.**

The heroes have arranged a bunch of trips where the i^{th} trip requires G gold bars to be transported from town A_i to B_i. **(Note that G is constant across all trips.)** Which is to say, in addition to the gold bars being used to pay for the tolls, an additional G have to be delivered to the destination town at the end of the trip. The heroes would like to minimise the amount of fuel used and would take the optimal route and carry the optimal number of gold bars to pay the tolls and minimise fuel usage. However, between trips the toll cost of certain roads may change and thus affect the fuel used in future trips.

Since, the heroes are busy preparing for battle, they do not have time to calculate their fuel usage and would like you to do so for them for each trip (this is a query operation). Keep in mind that between trips the toll cost of certain roads may change (this is an update operation). Given the correct order of events with the trips they are planning and the toll changes of roads, calculate their fuel consumption for each trip. As the result can be very large, you should give the answer modulo $10^9 + 7$.

Input

Your program must read from standard input.

The first line contains two integers N and G, the number of towns and the number gold bars each trip will transport.

$N - 1$ lines will follow. The i^{th} line contains 4 integers, A_i, B_i, D_i and T_i.

The next line contains a single integer Q representing the total number of trips and changes to tolls on the roads (i.e. the total number of query and update operations).

Q lines will follow. The i^{th} line will begin with an integer V_i:

- If $V_i = 0$, this represents a update operation; the line will contain 3 more integers X Y T, which indicates that the toll of the road that connects town X and Y is changed to T.
- If $V_i = 1$, this represents a query operation; the line will contain 2 more integers X Y, which indicates a trip from town X to town Y.

Output

Your program must print to standard output.

For each query operation, you should output a line containing a single integer — the minimum fuel used in the trip modulo $10^9 + 7$.

The result of each trip needs to be outputted in the same order as given in the input.

Implementation Note

As the input lengths may be very large, you are recommended to use C++ with fast input routines to solve this problem. The scientific committee does not have a solution written in Java or Python that can fully solve this problem.

C++ and Java source files containing fast input/output templates have been provided in the attachment. You are strongly recommended to use these templates.

If you are implementing your solution in Java, please name your file `Truck.java` and place your `main` function inside `class Truck`.

Subtasks

The maximum execution time on each instance is 2.0s, and the maximum memory usage on each instance is 512MiB. For all testcases, the input will satisfy the following bounds:

- $2 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq A_i, B_i \leq N$
- $1 \leq D_i, T_i, G \leq 10^9$

Your program will be tested on input instances that satisfy the following restrictions:

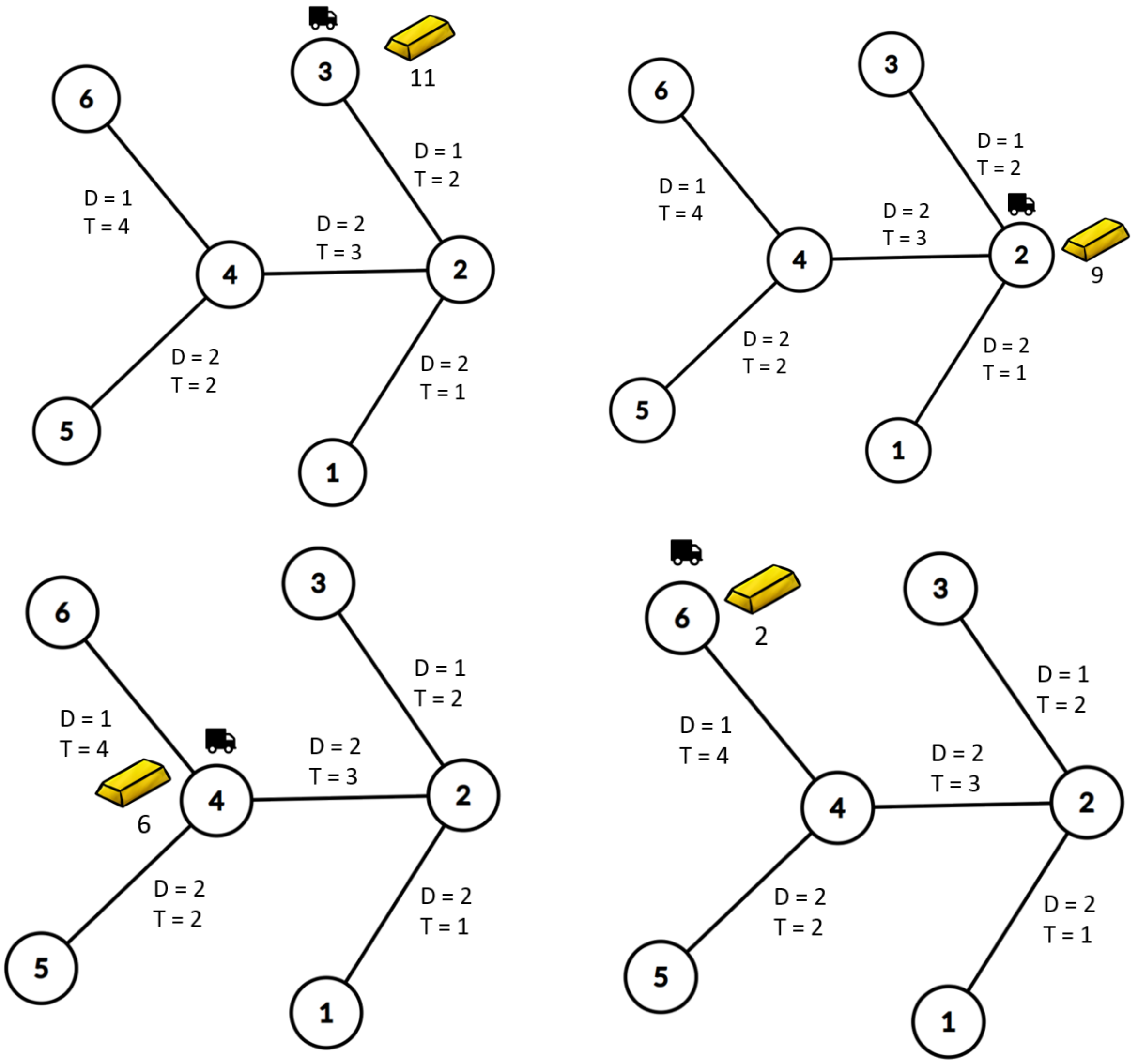
Subtask	**Marks**	V_i	**Additional Constraints**
1	5	$V_i = 1$	Each town has at most **two roads** connected to it $T_i = 0$
2	9		$T_i = 0$
3	12		$T_i = T_j$ for all $i \neq j$ $D_i = 1$
4	17		Each town has at most **two roads** connected to it
5	20	$0 \leq V_i \leq 1$	Each town has at most **two roads** connected to it
6	18		$N, Q \leq 5000$
7	19		-

Sample Testcase 1

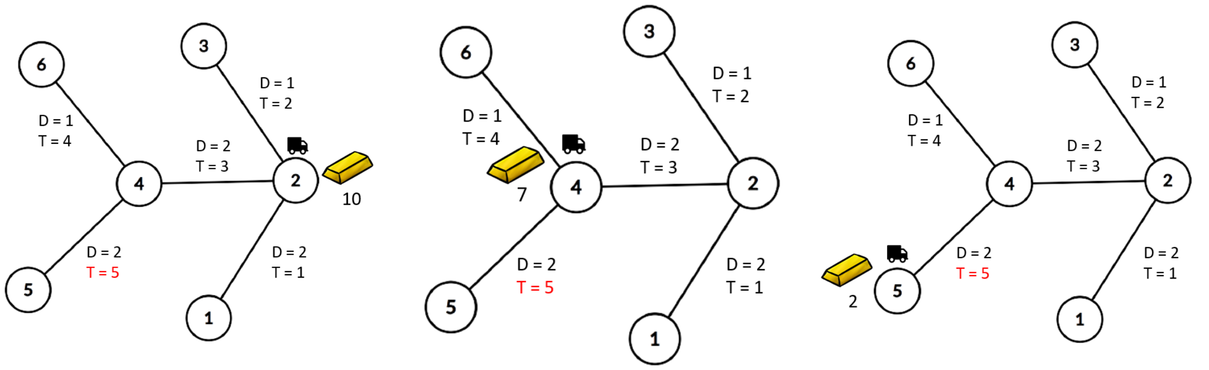
This testcase is valid for subtasks 6 and 7.

Input	Output
6 2 1 2 2 1 2 3 1 2 2 4 2 3 4 5 2 2 4 6 1 4 3 1 3 6 0 4 5 5 1 2 5	23 18

Sample Testcase 1 Explanation



The first query asks for a journey from node 3 to 6, transporting 2 gold bars. The truck will start at node 3 with 11 gold bars as it is the minimum number of gold bars required to end at node 6 with 2 gold bars. Travelling from node 3 to 2, we first pay 2 gold bars for the toll, ending up with 9 gold bars. Travelling a distance of 1 unit with 9 gold bars consumes 9 units of fuel. Similarly, from node 2 to 4 we have 6 gold bars, consuming 12 units of fuel and from node 4 to 6 we have 2 gold bars consuming 2 units of fuel. Totaling up to 9 + 12 + 2 = 23 units of fuel.



For the second query we change the toll on the road connecting 4 and 5 to 5. Then the third query asks for a journey from node 2 to 5 to transport 2 gold bars. Optimally, we begin at node 2 with 10 gold bars. To travel from node 2 to 4 we first spend 3 gold bars on the toll, to be left with 7 bars and then travel the road, consuming 14 units of fuel. Similarly, we travel on the road from node 4 to 5 with 2 gold bars consuming 4 units of fuel, totalling to 14 + 4 = 18 units of fuel.

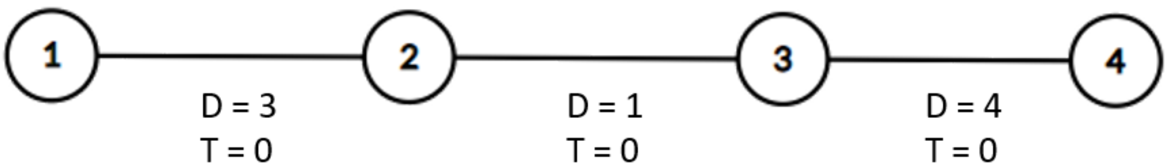
Sample Testcase 2

This testcase is valid for subtasks 1, 2, 4, 5, 6 and 7.

Input	Output

`4 3`
`1 2 3 0`
`2 3 1 0`
`3 4 4 0`
`1`
`1 1 4`
	`24`

Sample Testcase 2 Explanation



There are no tolls, hence the truck can just start will 3 gold bars and travel from node 1 to 4. Travelling a distance of 8 units, thus consuming $8 \times 3 = 24$ units of fuel.