Fire Nation Vaccine Distribution

Source: https://truckmagz.com/fitur-baru-truk-hino-untuk-segmen-kargo/

Description

In order to fight the COVID-19 that has hit the Fire Nation, the government is taking firm steps to produce a vaccine that is effective in eradicating this virus. Several months ago, the government produced, tested, and distributed the vaccine to several vaccine banks scattered throughout the Fire Nation. Currently, the government orders these cities to distribute vaccines to other cities that do not yet have vaccines.

In pursuing this effort, the government points to Dr. Watame as a coordinator. You are a renounced Software Engineer trusted by Dr. Watame to build a main algorithm in the vaccine distribution system. The algorithm to be made has to be effective, efficient, and 100% optimal. As a good and professional Software Engineer, help Dr. Watame to build this system.

The Fire Nation will have a configuration of roads in its country. This configuration depends on the type of the road. There are 3 types of roads, namely type 0, type 1, type 2. Roads type 0 and type 1 have no directional restrictions so they can be used both ways. While type 2 roads have directional restrictions. Roads with type 0 will have configuration ***A***, ***B***, and ***W***. That is, the road has a length of ***W***, and connects city A with city B (directly, you don't have to pass another city first). Roads with type 1 and 2 have the same configuration as ***A*** and ***B*** only, and both have no length so that when used the length is 0. Configurations that have been made can be deleted using **DELETE** according to the type of road. Not all cities are connected to other cities meaning there are cities in Fire Nation that are not connected to each other (there are no paths or a series of connecting roads). A summary of the road types mentioned can be seen in table 1.

Table 1. Summary of road types and properties

Road Type	One way/ two way	Length	Insert Configuration	Delete Configuration
Type 0	Two way	Has a length	`INSERT 0 A B W`	`DELETE 0 A B`
Type 1	Two way	Has no length (0)	`INSERT 1 A B`	`DELETE 1 A B`
Type 2	One way	Has no length (0)	`INSERT 2 A B`	`DELETE 2 A B`

Other than these configurations, the government will also give commands to these cities. Below are the types of commands that'll be given by the government.

- **`SHORTEST_PATH 0 A B`**
  To cut time, the government issues the **SHORTEST_PATH 0** command to find the shortest distribution path between city **A** and **B** by only using the road of type 0. **This command will output that shortest distance.** If the route from city **A** and **B** doesn't exist, the command will output **"-1"** (without quotation marks). **A and B are guaranteed to be different cities.**

- **`SHORTEST_PATH 1 A B`**
  To reduce time, the government issued an order **`SHORTEST_PATH 1`** for finding the shortest distribution path between city ***A*** and ***B***. The government allows truck drivers to pass any number of type 0 roads and type 1 roads a maximum of one time. **This command will generate the shortest distance.** If the route from city ***A*** to ***B*** is not available, the command will issue "**-1**" (without the quotes). **A and B are guaranteed to be different cities.**

- **`MIN_PATH A B`**
  The government wants to know the minimum number of roads to go if the vaccine delivery truck from city ***A*** wants to go to city ***B***. **This command will result in the minimum number of roads traveled.** If the route from city ***A*** to ***B*** is not available, the command will issue "**-1**" (without the quotes). **A and B are guaranteed to be different cities.**

- **`IS_CONNECTED A B`**
  The government wants to know whether city **A** has a path to city **B.** This command only uses roads of type 0. **This command will output "1" (without quotations) if city A has a path to reach city B. If not, this command will output "0" (without quotations) A and B are guaranteed to be different.**

- **`COUNT_CITY A M`**
  The government wants to know how much is the maximum number of cities reachable by a vaccine delivery truck when moving from city **A** with a maximum distance of **M**. This command only uses roads of type 0. **This command will output the maximum number of cities reachable including city A itself.**

- **`COUNT_CONNECTED`**
  The government would like to know how many provinces there are in the Fire nation. A province is defined as a collection of cities maximally connected with roads of type 0. **This command will output the amount of connected graphs from the configuration of roads in**

the Fire Nation at the time. *Note: Look at Discrete Mathematics 2 material regarding the concept of graph components (maximally connected subgraphs).*

- **`SIMULATE_WALK` *`A S`***
  Starting from city **A**, the government ordered trucks to follow special roads as many as **S** roads (guaranteed for each city, there is a maximum of one special road that goes out of city and into that city). The government wants to know if the truck is going as many as **S** steps, the truck will stop in what city. **This command will return the city where the truck has stopped.**

Input

The first line consists of two numbers, namely **N** and **Q** separated by a space.

N is a total of cities that will be created. As example, if N is 7, then the cities that will be created are city 0 up to city 6.

For each line in the next **Q** lines, each consist of one road configuration from Fire Nation or commands from the government

If the line contains a certain configuration. It will be preceded by **`INSERT/DELETE 0`** for paths of type 0, **`INSERT/DELETE 1`** for paths with type 1, or **`INSERT/DELETE 2`** for paths with type 2. Each configuration parameter will depend on parameters required for each configuration. For details about the parameters, refer to Table 1.

If the line consists of commands from the government described above, it will be started with a string followed by several parameters according to the command written. The first string indicates the type of query based on its abbreviation. In accordance with the type of query, the following are the parameters for each query.

- **`SHORTEST_PATH 0`** is followed by two positive integers **A** and **B** denoting the name of cities
- **`SHORTEST_PATH 1`** is followed by two positive integers **A** and **B** denoting the name of cities.
- **`MIN_PATH`** is followed by two positive integers **A** and **B** denoting the name of cities.
- **`IS_CONNECTED`** is followed by two positive integers **A** and **B** denoting the name of cities.
- **`COUNT_CITY`** is followed by a positive integer **A** denoting a starting point city and a positive integer **M** that denotes the maximum distance given.
- **`SIMULATE_WALK`** is followed by an integer **A** denoting the city starting point and a positive integer **S** denoting the maximum distance given.
- A string **COUNT_CONNECTED** that does the query processing counting the number of provinces in the Fire Nation.

Output

Output consists of many lines of commands created. Row *`i`* contains a number which is the answer to the *`i`*-th query.

Constraints

- $1 \leq N \leq 10^4$
- $1 \leq Q \leq 10^4$
- $0 \leq A, B \leq N-1$
- $1 \leq W \leq 10^4$
- $1 \leq M \leq 10^9$
- $1 \leq S \leq 10^{18}$
- A ≠ B

All road lengths of type 0 are guaranteed to be positive integers.

Input Example 1

```
7 22
INSERT 0 1 2 3
INSERT 0 1 3 6
INSERT 0 2 3 1
INSERT 0 3 4 20
INSERT 0 5 6 11
SHORTEST_PATH 0 1 2
SHORTEST_PATH 0 1 3
SHORTEST_PATH 0 4 1
MIN_PATH 1 4
MIN_PATH 6 5
IS_CONNECTED 4 2
IS_CONNECTED 1 5
IS_CONNECTED 0 6
COUNT_CITY 2 10
COUNT_CITY 2 100
COUNT_CONNECTED
DELETE 0 1 2
SHORTEST_PATH 0 1 3
IS_CONNECTED 1 2
COUNT_CONNECTED
DELETE 0 1 3
COUNT_CONNECTED
```

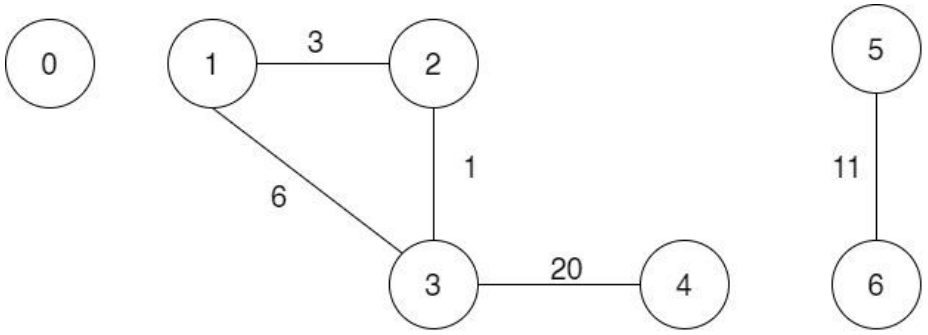
Output Example

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

```
6
1
3
4
```

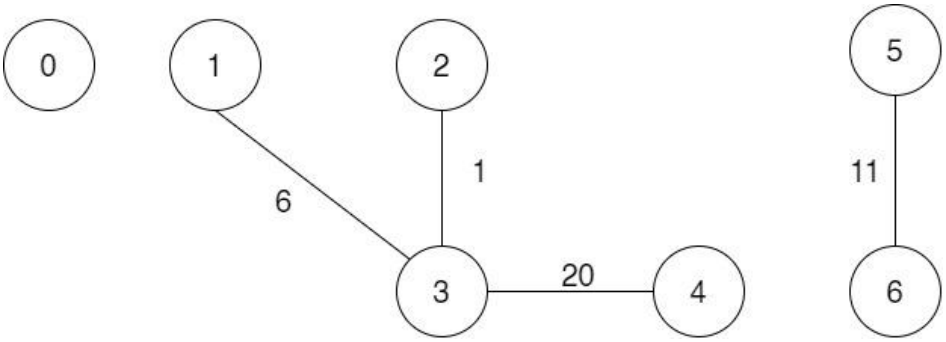
Explanation

Graph illustration after **`INSERT 0 5 6 11`**:



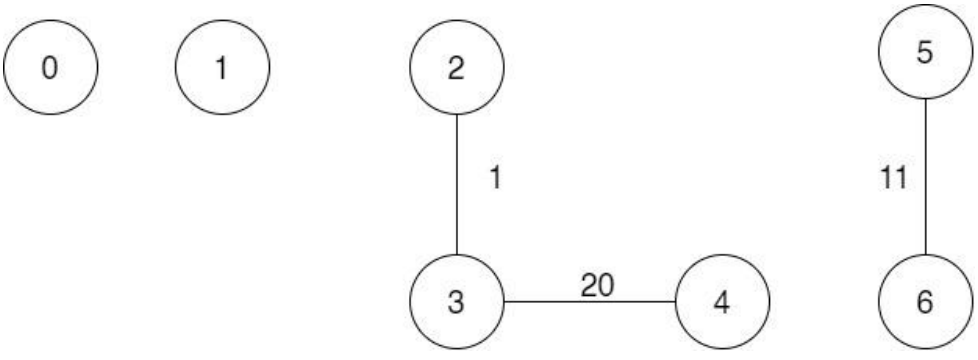
Graph 1.1

Graph illustration after **`DELETE 0 1 2`**:



Graph 1.2

Graph illustration after **`DELETE 0 1 3`**:



Graph 1.3

Input	Output	Explanation

SHORTEST_PATH 0 1 2	3	This query requests the shortest distance from city 1 to city 2 by using roads of type 0 only. It can be seen in graph 1.1 that the shortest distance is through path 1-2 which has a total distance of 3
SHORTEST_PATH 0 1 3	4	This query requests the shortest distance from city 1 to city 3 using roads of type 0 only. It can be seen in graph 1.1 that the shortest distance is through path 1-2-3 which has a total distance of 3 + 1 = 4
SHORTEST_PATH 0 4 1	24	This query requests the shortest distance from city 4 to city 1 using roads of type 0 only. It can be seen in graph 1.1 that the shortest distance is through the path 4-3-2-1 which has a total distance of 20 + 3 + 1 = 24
MIN_PATH 1 4	2	This query asks for a path with a minimum number of type 0 roads from city 1 to city 4. It can be seen in graph 1.1 that the optimal path is 1-3-4 with as many as 2 roads.
MIN_PATH 6 5	1	This query requests a path with a minimum number of roads of type 0 from city 6 to city 5. It can be seen in graph 1.1 that the optimal path is 6-5 with 1 as many roads.
IS_CONNECTED 4 2	1	This query asks for the connection between city 4 and city 2. It can be seen in Graph 1.1, because there is a path that uses road type 0 between the two cities, the output is 1.
IS_CONNECTED 1 5	0	This query asks for the connection between city 1 and city 5. It can be seen in Graph 1.1, because there is no path using type 0 roads between the two cities, we output 0.
IS_CONNECTED 0 6	0	This query asks for the connection between city 0 and city 6. It can be seen in Graph 1.1, because there is no path that uses roads of type 0 between the two cities, we output 0.
COUNT_CITY 2 10	3	This query asks how many cities can be reached by city 2 using only type 0 roads with a maximum distance of 10 (including city 2 itself). It can be seen in graph 1.1, the answer is cities 1, 2, and 3.The output is the number of cities, namely 3.
COUNT_CITY 2 100	4	This query asks how many cities can be reached by city 2 using only type 0 roads with a maximum distance of 100 (including city 2 itself). It can be seen in graph 1.1, the answer is cities 1, 2, 3, and 4. The output is the number of cities, namely 4.

COUNT_CONNECTED	3	This query asks us to count the number of provinces in the country of fire. It can be seen in graph 1.1 that the answer is 3, namely {0}, {1, 2, 3, 4}, and {5, 6}
SHORTEST_PATH 0 1 3	6	This query requests the shortest distance from city 1 to city 3 using roads of type 0 only. It can be seen in graph 1.2 that the shortest distance is through path 1-3 which has a total distance of 6
IS_CONNECTED 1 2	1	This query asks for the connection between city 1 and city 2. It can be seen in Graph 1.2, because there is a path that uses road type 0 between the two cities, we output 1.
COUNT_CONNECTED	3	This query asks us to count the number of provinces in the country of fire. It can be seen in graph 1.2 that the answer is 3, namely {0}, {1, 2, 3, 4}, and {5, 6}
COUNT_CONNECTED	4	This query asks us to count the number of provinces in the country of fire. It can be seen in graph 1.3 that the answer is 4, namely {0}, {1}, {2, 3, 4}, and {5, 6}

Input Example 2

```
7 18
INSERT 0 1 2 3
INSERT 0 1 3 6
INSERT 0 2 3 1
INSERT 0 3 4 20
INSERT 0 4 5 1234
INSERT 0 5 6 11
INSERT 1 4 5
INSERT 1 6 3
INSERT 2 1 3
INSERT 2 3 4
INSERT 2 4 5
INSERT 2 5 1
SHORTEST_PATH 0 1 5
SHORTEST_PATH 0 1 6
SHORTEST_PATH 1 1 5
SHORTEST_PATH 1 1 6
SIMULATE_WALK 1 6
SIMULATE_WALK 1 1000000000000
```

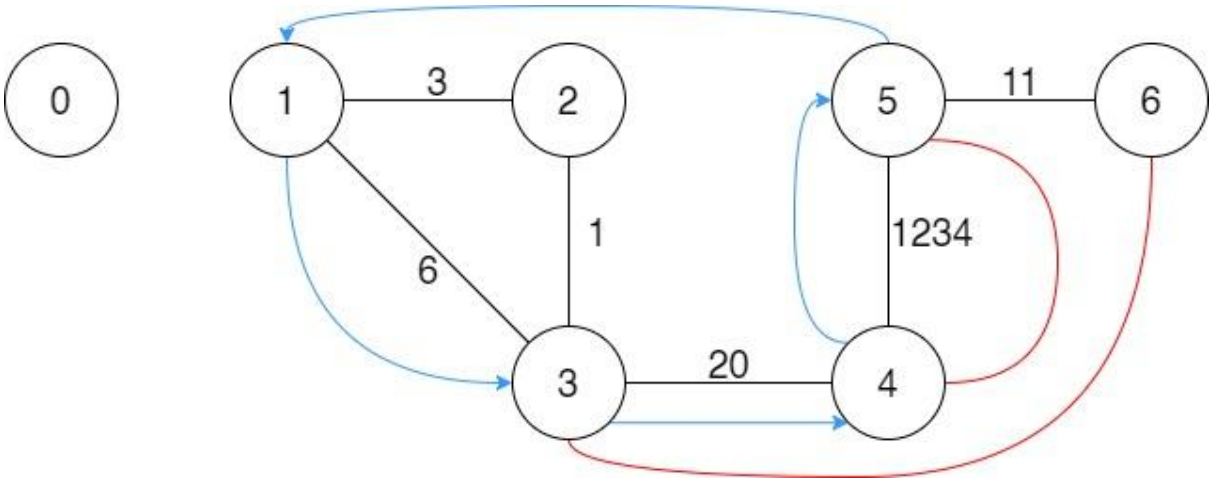
Output Example

```
1258
1269
```

```
15
4
4
1
```

Explanation

Graph illustration after **`INSERT 2 5 1`**:



Graph 2.1

Notes:

- Road BLACK is road type 0
- Road RED is road type 1
- Road BLUE is road type 2

Input	Output	Explanation
`SHORTEST_PATH 0 1 5`	1258	This query requests the shortest distance from city 1 to city 5 by using roads of type 0 only. It can be seen in graph 2.1 that the shortest distance is through the 1-2-3-4-5 path which has a total distance of 3 + 1 + 20 + 1234 = 1258
`SHORTEST_PATH 0 1 6`	1269	This query requests the shortest distance from city 1 to city 6 by using roads of type 0 only. It can be seen in graph 2.1 that the shortest distance is through the path 1-2-3-4-5-6 which has a total distance of 3 + 1 + 20 + 1234 + 11 = 1269
`SHORTEST_PATH 1 1 5`	15	This query requests the shortest distance from city 1 to city 5 by using any number of type 0 roads and type 1 roads a maximum of 1 times. It can be seen in graph 2.1, the optimal path is 1-2-3 <> 6-5 (<> is a type 1 road) which has a total distance of 3 + 1 + 11 = 15
`SHORTEST_PATH 1 1 6`	4	This query requests the shortest distance from city

		1 to city 6 by using any number of type 0 roads and type 1 roads a maximum of 1 times. It can be seen in graph 2.1, the optimal path is 1-2-3 <> 6 (<> is a type 1 road) which has a total distance of 3 + 1 = 4
SIMULATE_WALK 1 6	4	This query asks us to simulate a truck trip starting from city 1 using only type 2 roads for 6 steps. The resulting path is 1-3-4-5-1-3-4. The output is the city where the truck stops which is 4
SIMULATE_WALK 1 1000000000000	1	This query asks us to simulate a truck trip starting from city 1 using only type 2 roads of 1000,000,000,000 steps. The output is the city where the truck stops, namely 1.

Input Example 3

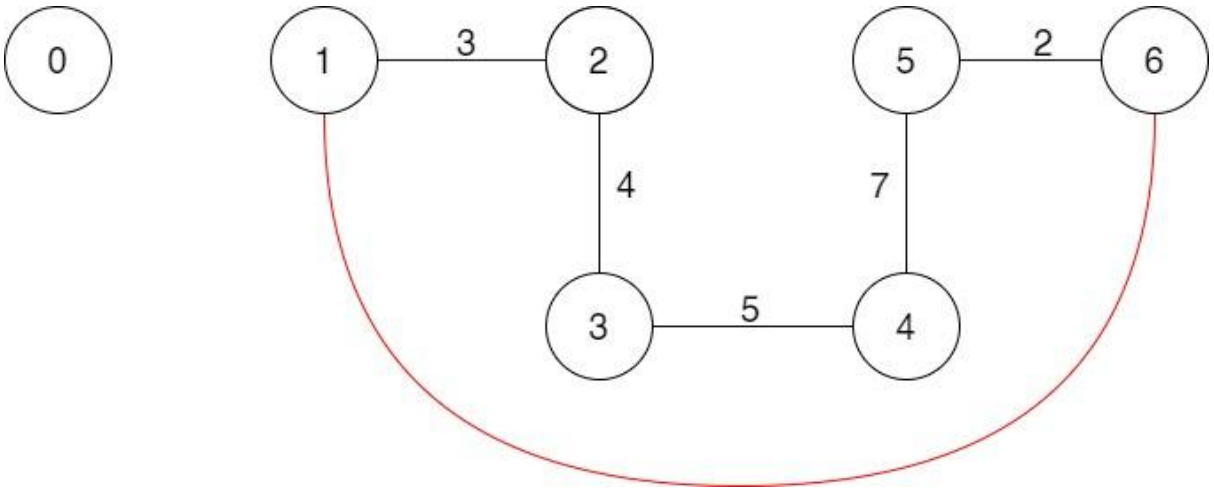
```
7 10
INSERT 0 1 2 3
INSERT 0 2 3 4
INSERT 0 3 4 5
INSERT 0 4 5 7
INSERT 0 5 6 2
INSERT 1 1 6
SHORTEST_PATH 0 3 4
SHORTEST_PATH 1 3 4
SHORTEST_PATH 0 1 6
SHORTEST_PATH 1 1 6
```

Output Example

```
5
5
21
0
```

Explanation

Graph illustration after INSERT 1 1 6:



Graph 3.1

Notes:

- Road BLACK is road type 0
- Road RED is road type 1

Input	Output	Explanation
SHORTEST_PATH 0 3 4	5	This query requests the shortest distance from city 3 to city 4 by using roads of type 0 only. It can be seen in graph 3.1 that the shortest distance is through path 3-4 which has a total distance of 5.
SHORTEST_PATH 1 3 4	5	This query requests the shortest distance from city 3 to city 4 by using any number of type 0 roads and type 1 roads a maximum of 1 times. It can be seen in graph 3.1, the optimal path is 3-4 which has a total distance of 5
SHORTEST_PATH 0 1 6	21	This query requests the shortest distance from city 1 to city 6 by using roads of type 0 only. It can be seen in Graph 3.1 that the shortest distance is through the path 1-2-3-4-5-6 which has a total distance of 3 + 4 + 5 + 7 + 2 = 21
SHORTEST_PATH 1 1 6	0	This query requests the shortest distance from city 1 to city 6 by using any number of type 0 roads and type 1 roads a maximum of 1 times. It can be seen in graph 3.1, the optimal path is 1 <> 6 (<> is a type 1 road) which has a total distance of 0

Test Case Distribution

Query	Test Case

SHORTEST_PATH 0	1 - 6
SHORTEST_PATH 1	7 - 12
MIN_PATH	13 - 18
IS_CONNECTED	19 - 24
COUNT_CITY	25 - 30
SIMULATE_WALK	31 - 36
Mixed Queries	37 - 50