Harry Potter and Enchanted Even Paths

The Gryffindor common room clock chimes past lights out, but Harry isn't in bed. He's on a mission. Fred and George Weasley's greatest prank yet —a dazzling display of Weasleys' Wildfire Whiz-bangs—depends on him retrieving their last batch of fireworks before Filch finds them. To do this successfully, the Weasley twins have given the Marauder's Map to Harry.

The Marauder's Map reveals Hogwarts as a **simple, undirected graph**: rooms as *nodes*, secret passages as *edges*, where *each passage has a travel time associated with it*. Harry wants to know the shortest path to the fireworks, but that's not all. Ever since his last encounter with Voldemort, Harry has a rather odd fear of paths with an odd number of passages. Therefore, he must use the **shortest path to the fireworks while ensuring it contains an even number of passages.**

To navigate efficiently, Harry recalls Hermione's lessons on data structures and algorithms. Since Hermione taught him about **min-priority queue**, he is comfortable only with those. However, STL priority queue was implemented by Professor Snape, and he always has information about which student uses his spells (and implementations). And Harry definitely doesn't want the professor to learn about his mischief. Therefore, he needs to implement a **generic minimum priority queue** first, to find the shortest path.

For more details, please refer to the document here

Input Format

The first line of each testcase contains two integers: n and m, which represent the number of rooms and number of bidirectional passages respectively.

Following n lines contain n unique room IDs. Each room ID is an alphanumeric string which corresponds to a unique room.

Following m lines contain one edge each in the form a_i, b_i, w_i, where a_i and b_i represent the room IDs and w_i represents the time it takes to traverse this passage.

The last line contains two room IDs: room ID of the source and room ID of the destination.

Constraints

$(2 \leq n \leq 1 \times 10^5, 1 \leq m \leq 1 \times 10^5)$

Length of a room ID is atmost 100 characters

Edge weights are positive. Total path weight between any two vertices will not exceed 2×10^9.

Output Format

Print a single integer, representing the minimum time required to traverse an even-length path from the source to the sink.

If such a path does not exist, print -1.

Sample Input 0

```
2 1
DarkArtsClassroom
ForbiddenForest
DarkArtsClassroom ForbiddenForest 5
ForbiddenForest DarkArtsClassroom
```

Sample Output 0

```
-1
```

Explanation 0

In this test case, there are two locations—DarkArtsClassroom and ForbiddenForest—connected by a single passage with a travel time of 5. Since the only route between them consists of one edge (an odd number), it doesn't satisfy the requirement for an even-length path. As a result, there's no valid way for Harry to travel from DarkArtsClassroom to ForbiddenForest, so the answer is -1.

Sample Input 1

```
3 3
GreatHall
AstronomyTower
RoomOfRequirement
GreatHall AstronomyTower 3
AstronomyTower RoomOfRequirement 2
GreatHall RoomOfRequirement 2
GreatHall AstronomyTower
```

Sample Output 1

```
4
```

Explanation 1

In this test case, three locations—GreatHall, AstronomyTower, and RoomOfRequirement—are connected by three edges: GreatHall–AstronomyTower (3), AstronomyTower–RoomOfRequirement (2), and GreatHall–RoomOfRequirement (2). The direct path from GreatHall to AstronomyTower is the shortest at 3, but since it uses one edge (odd), it’s not valid. Instead, the shortest even-length path through RoomOfRequirement which needs total of 4 minutes to traverse.

Sample Input 2

```
4 6
GryffindorTower
SlytherinDungeon
RavenclawTower
```

```
HufflepuffBasement
GryffindorTower SlytherinDungeon 119
GryffindorTower RavenclawTower 130
GryffindorTower HufflepuffBasement 170
SlytherinDungeon RavenclawTower 121
SlytherinDungeon HufflepuffBasement 145
RavenclawTower HufflepuffBasement 200
GryffindorTower SlytherinDungeon
```

Sample Output 2

```
251
```