D. Exploration plan

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The competitors of Bubble Cup X gathered after the competition and discussed what is the best way to get to know the host country and its cities.

After exploring the map of Serbia for a while, the competitors came up with the following facts: the country has V cities which are indexed with numbers from 1 to V, and there are E bi-directional roads that connect the cites. Each road has a weight (the time needed to cross that road). There are N teams at the Bubble Cup and the competitors came up with the following plan: each of the N teams will start their journey in one of the V cities, and some of the teams share the starting position.

They want to find the shortest time T, such that every team can move in these T minutes, and the number of different cities they end up in is at least K (because they will only get to know the cities they end up in). A team doesn't have to be on the move all the time, if they like it in a particular city, they can stay there and wait for the time to pass.

Please help the competitors to determine the shortest time T so it's possible for them to end up in at least K different cities or print `-1` if that is impossible no matter how they move.

Note that there can exist multiple roads between some cities.

Input

The first line contains four integers: V, E, N and K ($1 \leq V \leq 600,\ 1 \leq E \leq 20000,\ 1 \leq N \leq min(V, 200),\ 1 \leq K \leq N$), number of cities, number of roads, number of teams and the smallest number of different cities they need to end up in, respectively.

The second line contains N integers, the cities where the teams start their journey.

Next E lines contain information about the roads in following format: $A_i\ B_i\ T_i$ ($1 \leq A_i, B_i \leq V,\ 1 \leq T_i \leq 10000$), which means that there is a road connecting cities A_i and B_i, and you need T_i minutes to cross that road.

Output

Output a single integer that represents the minimal time the teams can move for, such that they end up in at least K different cities or output `-1` if there is no solution.

If the solution exists, result will be no greater than 1731311.

Example

input
6 7 5 4 5 5 2 2 5 1 3 3 1 5 2 1 6 5 2 5 4 2 6 7 3 4 11 3 5 3

output
3

Note

Three teams start from city 5, and two teams start from city 2. If they agree to move for 3 minutes, one possible situation would be the following: Two teams in city 2, one team in city 5, one team in city 3 , and one team in city 1. And we see that there are four different cities the teams end their journey at.