Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *Planet Express.*

It is year 3000 and you are an employee of the Planet Express delivery company. A customer from planet *Omicron Persei 8* just ordered one yottagram of dark matter using the "same-second" delivery option and you have been given the task of (quickly) planning the delivery.

Planet Express stores dark matter in $k > 0$ warehouses scattered on planets across the galaxy, and you can initiate the delivery process by choosing any of the warehouses and activating one of its robotic delivery agents.

The galaxy can be modelled as a weighted directed graph $G = (V, E)$, where $V = \{0, \dots, n-1\}$, and $|E| = m$. Each vertex of G represents a planet while an edge (u, v) of weight c means that it is possible to *travel* from planet u to planet v in c microseconds.[1] The k warehouses reside on planets $0, \dots, k-1$ while Omicron Persei 8 always corresponds to vertex $n-1$.

In addition, η of the planets are also part of a convenient *teleportation network*. Unfortunately, the teleportation technology still requires the source and destination planets to be reachable from one-another by regular means of travel. More precisely, we say that two *distinct* planets u and v **that are *both* part of the teleportation network** are *linked* if it is possible to reach v from u and vice-versa without using the teleportation network. An agent on planet u can *teleport* to the location of any another planet v that is linked with u in just $t(u)$ microseconds, where $t(u)$ is the total number of planets that are linked with u.

Once activated, the delivery agent will deliver the dark matter to the customer in the shortest possible time, travelling between planets and/or using the teleportation network.

Your task is to determine whether it is possible for Planet Express to complete the delivery within the advertised time of 1 second and, if that is the case, the shortest amount of time required for the delivery. You can assume that the time needed to travel between two locations on the same planet is negligible.

Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line contains the number T of test-cases. The first line of each test case contains the four integers n, m, k, and η, separated by a space. The second line of the test case contains η distinct integers corresponding to indices the planets (i.e., vertices of G) that are part of the teleportation network. Each of the following m lines describes one the edges of G and contains three integers u v and $c_{u,v}$ to encode the fact that G contains a directed edge (u, v) of weight c.

Output. The output consists of one line for each test case. If in the i-th test case the dark matter cannot be delivered in at most 1 second, then the i-th line of the output should contain the string "`NO`". Otherwise, the i-th line of the output should contain a single integer that denotes the smallest number of microseconds needed to complete the delivery (i.e., the time needed be the delivery agent to reach Omicron Persei 8 once the best warehouse has been selected).

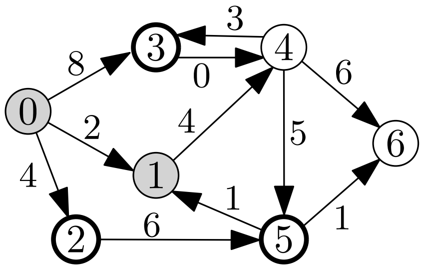
Assumptions. $1 \leq T \leq 10$; $1 \leq n \leq 2^{16}$; $1 \leq n \leq 2^{18}$; $0 \leq k \leq n$;
$\forall (u, v) \in E$ $1 \leq c_{u,v} \leq 2^{22}$.

(Continues on the next page)

[1] A microsecond is equal to 10^{-6} seconds. Spaceships are fast!

Example.

In the following example $n = 7$, $m = 11$, $k = 2$, and $\eta = 3$. Vertices that represent warehouses are shown in gray. Vertices that are part of the teleportation network are shown in bold.



An optimal solution delivers the black matter from warehouse 1. The required delivery time is $4+3+1+1 = 9$ microseconds and is attained by traversing the sequence of vertices $\langle 1, 4, 3, 5, 6 \rangle$. Notice that the teleportation network has been used to travel from planet 3 to planet 5.

Input (the first test-case corresponds to the example above):

```
2
7 11 2 3
2 3 5
0 1 2
0 2 4
0 3 8
1 4 4
2 5 6
3 4 0
4 3 3
4 5 5
4 6 6
5 1 1
5 6 1
5 6 2 2
2 3
0 1 5
0 3 1000001
1 2 999500
2 0 0
3 0 100
3 4 500
```

Output:

```
9
NO
```

Requirements. Your algorithm should require $O(m + n \log n)$ time (with reasonable hidden constants).

Notes. A reasonable implementation should not require more than 1 second for each input file.