Task: SUS
System for Universal Signalling

AACPP SuSe 2024 | Round 2 | Memory: 64MiB (Java: 256MiB) | 2024.05.07 – 2024.05.28

Managing large distributed systems is quite a challenge. Byteman is now in charge of the System for Universal Signalling, so they can attest to that.

SUS is architected as a connected network of n nodes and m direct links between pairs of nodes. To ensure high availability, k of the nodes are designated as *key nodes* responsible for keeping the state of the system consistent.

While most of the network's operations are already working well, there is one scenario that requires close attention to ensure maximum efficiency – *signal propagation*. The procedure is defined as follows:

- Any of the n nodes issues a signal.
- Until all key nodes have recorded the signal:
  - When a regular node receives the signal can propagate it to **at most one** other node directly linked to it. It then forgets about the signal.
  - A *key* node that has received the signal records it and has the authority to propagate it to **any number** of nodes directly linked to it.

Each issued signal is unique and does not affect the other signals. While correctness is necessary, this operation happens millions of times per minute and its carbon footprint cannot be ignored. The links between nodes are of different lengths and use more or less efficient communication technologies. The carbon footprint of one activation of the link between nodes i and j is estimated as $c_{i,j}$. The total cost of a single signal propagation operation is the sum of carbon footprints of all activated links.

Help Byteman with optimising SUS. For each signal propagation operation calculate the minimal carbon footprint cost required to propagate it to every key node.

Input

In the first line of standard input there are four integers n, m, k, q, in order: the number of nodes in the SUS network, the number of links between them, the number of key nodes, the number of signal propagation queries. There are at least 2 nodes in the network.

The next line contains k integers describing which nodes are the key nodes. They are all between 1 and n, and are unique. There is at least one key node.

The next m lines contains descriptions of links in the network. Each line contains three integers, $i, j, c_{i,j}$, designating a two-way link between nodes i and j with a carbon footprint of $c_{i,j}$. Each unordered pair of nodes will appear at most once, and there are no links between a node and itself.

The final q lines contain signal propagation queries. Each line contains one integer q_i between 1 and n describing a signal propagation operation starting from the node q_i.

Output

Your program should write q lines to standard output. The i-th line should contain the minimal sum of carbon footprint costs of all links that need to be activated in order to complete the signal propagation from node q_i.

Example

For the input:

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

the correct output is:

```
10
8
10
```

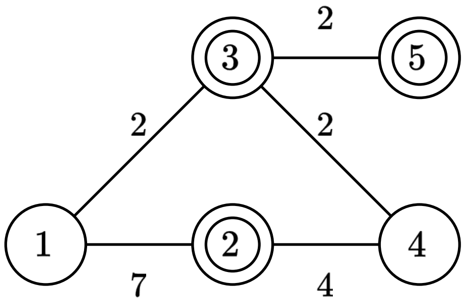
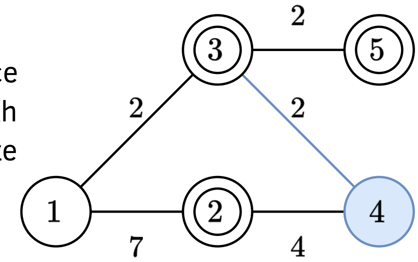
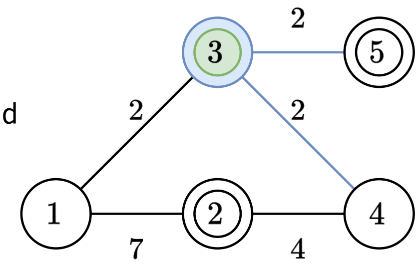


Figure 1: Visual representation of the network. Key nodes are marked with doubled outlines.

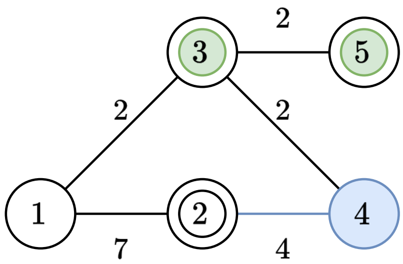
Let us elaborate on the last query, starting at node (4). Since (4) is not a key node it cannot propagate the signal to both neighbouring key nodes. The optimal choice is to propagate to (3), paying a cost of 2.



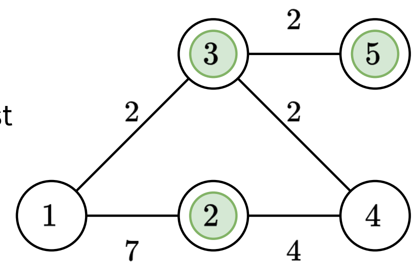
Since (3) is a key node, it can choose to propagate to (5) and also back to (4), paying 2 for each.



Now (4) can relay the signal to (2) paying 4.



And with that the signal reached all key nodes. The total cost comes down to $2 + (2 + 2) + 4 = 10$.



Additional examples

The following initial tests are also available:

- `0b` – sample for Subtask 1, $n = 10$, $m = 10$, $k = 2$, $q = 4$, all costs 1;
- `0c` – sample for Subtask 2, cycle on $n = 10^3$ vertices, $k = 10$, equidistant key nodes;
- `0d` – sample for Subtasks 3 and 4, binary tree on $n = 10^5$ vertices, $k = 128$, key nodes span all levels;
- `0e` – big test, $n = 10^5$, $m = 499985$, $k = 100$, $q = 10^5$, vertex v has edges to vertices $v + i$ for $1 \leq i \leq 5$ (as long as those exist) and the weights are squares of i.

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

Subtask	Limits	Points
1.	$2 \leq n \leq 10$, $1 \leq m \leq 20$, $1 \leq k \leq 10$, $1 \leq q \leq 10$	2
2.	$2 \leq n \leq 10^3$, $1 \leq m \leq 5 \cdot 10^5$, $1 \leq k \leq 10$, $1 \leq q \leq 20$	1
3.	$2 \leq n \leq 10^5$, $1 \leq m \leq 5 \cdot 10^5$, $1 \leq k \leq 128$, $1 \leq q \leq 20$	2
4.	$2 \leq n \leq 10^5$, $m = n - 1$, $1 \leq k \leq 128$, $1 \leq q \leq 10^5$	2
5.	$2 \leq n \leq 10^5$, $1 \leq m \leq 5 \cdot 10^5$, $1 \leq k \leq 128$, $1 \leq q \leq 10^5$	3