CRYPTO

Problem

Vim hacker Granny now wants to get in on the cryptocurrency hype. However, she hasn't really figured out what decentralization means yet. In exchange for some her new cryptocurrency, you have agreed to be the central planner of her not-yet-decentralized network. Your job is to determine the best way to connect the users of the cryptocurrency to Granny's servers. For QQQ different values qiq_iqi, your task is to determine the maximum bandwidth the user with the qiq_iqi-th highest bandwidth can have.

The network you are interested in can be described as a weighted undirected graph of nnn vertices and mmm edges. The vertices, numbered 111 through nnn, represent network endpoints. The edges represent connections, with the weight of each edge corresponding to the quality of the connection. Traffic can be routed along any path in this network; hence the bandwidth between two vertices is the weight of the bottleneck shortest path between the vertices. (The bottleneck shortest path is the path such that the weight of the minimum weight edge is maximized.)

Among the network endpoints, a set of kkk endpoints corresponds to the servers, the jjj-th of which has the capacity to support cjc_jcj users. A disjoint set of ttt endpoints corresponds to the users. Given a description of the network, along with the locations of the servers and users, your task is to answer the QQQ queries. For the iii-th query, you should compute the maximum bib_ibi such that it is possible to assign each user to a server so that each of the qiq_iqi best user-server connections have bandwidth at least bib_ibi and no server is assigned more users than its capacity cjc_jcj.

Constraints

There is one sample input, worth 0 points. (See "Sample Input" below.) This test serves to verify that the server is assessing your code correctly, in the event that all of the other test cases are reported as Incorrect Output.

For all other test cases, worth a total of 100 points, it is guaranteed that 2≤n≤1052\le n\le 10^52≤n≤105, m≤2·105m\le 2\cdot 10^5m≤2·105, and 1≤Q≤1051\le Q\le 10^51≤Q≤105. These test cases are further divided into four batches, with the following additional constraints:

- in the first batch, worth 10 points, it is guaranteed that n≤15n\le 15n≤15, m≤100m\le 100m≤100, Q=1Q=1Q=1, q1=tq_1 = tq1=t, and k=1k=1k=1;
- in the second batch, worth 20 points, it is guaranteed that n≤15n\le 15n≤15, m≤100m\le 100m≤100, Q=1Q=1Q=1, and q1=tq_1 = tq1=t;
- in the third batch, wroth 30 points, it is guaranteed that n≤1000n\le 1000n≤1000, m≤104m\le 10^4m≤104, Q=1Q=1Q=1, and q1=tq_1 = tq1=t;
- the fourth batch, with no additional constraints, is worth the remaining 40 points.

Finally, it is guaranteed for all input files that the network graph is connected, that no vertex is both a server and a user, and that at least one valid assignment of users to servers exists.

Time Limit

The time limit is 100ms for each test case in the first batch, 200ms for each test case in the second and third batches, and 1000ms for each test case in the fourth batch.

Input Format

Each input consists of m+7m+7m+7 lines as follows:

- line 111 consists of two space-separated integers, nnn and mmm;
- lines 222 through m+1m+1m+1 each consist of three space-separated integers aia_iai, bib_ibi, and wiw_iwi, indicating an edge between vertices aia_iai and bib_ibi with weight wiw_iwi, 1≤wi≤1061\le w_i\le 10^61≤wi≤106;
- line m+2m+2m+2 consists of two space-separated integers, kkk and ttt, 1≤k,t<n1\le k, t < n1≤k,t<n;
- line m+3m+3m+3 consists of kkk, distinct space-separated integers s1,…,sks_1,\ldots,s_ks1,…,sk, 1≤sj≤n1\le s_j\le n1≤sj≤n, describing the locations of the servers;
- line m+4m+4m+4 consists of kkk space-separated integers c1,…,ckc_1,\ldots,c_kc1,…,ck, 1≤cj≤1061\le c_j\le 10^61≤cj≤106, with cjc_jcj describing the capacity of server sjs_jsj, i.e., the number of users it can support;
- line m+5m+5m+5 consists of ttt distinct space-separated integers u1,…,utu_1,\ldots,u_tu1,…,ut, 1≤uj≤n1\le u_j\le n1≤uj≤n, describing the locations of the users.
- line m+6m+6m+6 consists of one integer, the number of queries QQQ;
- line m+7m+7m+7 consists of QQQ space-separated integers q1,…,qQq_1,\ldots,q_Qq1,…,qQ, 1≤qQ≤t1\le q_Q\le t1≤qQ≤t, describing the queries.

Output Format

Output QQQ lines, with the kkk-th line containing the answer to the kkk-th query.

Sample Cases

Sample Input


```
6 6
1 2 2
2 3 6
3 4 8
4 5 5
5 6 4
6 1 5
2 4
3 6
3 3
1 2 4 5
1
4
```

Sample Output

```
5
```

Details

The optimal solution is to assign the users at 222, 444, and 555 to the server at 333 and the user at 111 to the server at 666. The lowest

bandwidth connections are those of the users at 111 and 555, each of which has bandwidth 555.

View submissions

Test cases

Input	Output	Points	Timeout
6 6 1 2 2 2 3 6 3 4 8 4 5 5 5 6 4 6 1 5 2 4 3 6 3 3 1 2 4 5 1 4	5	0	100 ms
Hidden	*Hidden*	5	100 ms
Hidden	*Hidden*	5	100 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	200 ms
Hidden	*Hidden*	10	1000 ms
Hidden	*Hidden*	10	1000 ms
Hidden	*Hidden*	10	1000 ms
Hidden	*Hidden*	10	1000 ms

Download

Inspired by the "Ultra Cool Programming Contest Control Centre" by Sonny Chan.
Modified for CS 124 by Neal Wu, with design help from Martin Camacho.
Further refined by Nikhil Benesch.