WikiNet

Wikipedia has articles on different subjects (categories) such as Sports, Education, History, Sciences, etc. Each article has links to its supporting articles. The referred articles may or may not have links back to the referring article. Let us assume that each link takes some time to load based on the contents of the page.

Given the network (or graph) **G**, of Wikipedia articles, consisting of articles (nodes), links associated with each article (edges) and time taken to load each link (edge weights), implement functions to accomplish the following tasks:

min-time(G, s): Given an article **s**, find the least time required, starting from **s**, to load all the other articles *(via direct links or through a sequence of links (path))*. If an article is unreachable from **s** then the least time is **-1**.

update-network(G, s, m): If an article **t** is unreachable from **s** or takes longer than **m**-minutes (time limit) to load from **s**, then update **G** with a link from **s** to **t** which takes **m**-minutes to load.

Top-k(G, c, k): In the updated **G**, print the top **k** articles that are most referred by the articles of category **c**. In case of a tie, consider them in the increasing order of article number. If there are less than **k** articles referred by articles of category **c**, then report all referred articles in descending order of their reference counts.

pair-dist(G, c1, c2): In the updated **G**, print the least time path (sequence of articles) from all articles of category **c1** (order by article number) to all the articles of category **c2**.

In case there are more than one least-time paths between any pair of articles, report the path which has the least article id at the first point of difference, starting at the source.

For example, if there are two shortest paths: **5 --> 7 --> 9 --> 8** and **5 --> 7--> 4 --> 3--> 8** from **5** to **8**, then we need to report the path **5 --> 7--> 4 --> 3--> 8**, since at the first point of difference (from the source) $4 < 9$.

Note: If an article is not reachable from another article, then print -1.

Notation

N - Number of articles in Wikipedia. Each named as integer from 1 to N.

L - Number of links between the articles.

C - Number of categories of articles. Each named as integer from 1 to C.

$t(u, v)$ - Time to load link from article u to article v. All load times are floating-point numbers.

s - Source article.

k - Top-k referred articles, referred by the articles of category of source article s.

Input Format

$N\ L$

N space separated integers specifying categories (any one of $1..C$) for $1..N$ articles.

$x_1\ \ y_1\ \ t_1$ // L lines indicating directed links from x_i to y_i with time t_i.

$x_2\ \ y_2\ \ t_2$

...

x_L y_L t_L

sssp <s> // single source shortest path or the *min-time()* task as defined.

update <s> <m> // *update()* task as defined.

top-k <c> <k> // top k articles referred by articles of category c.

apsp <$c1$> <$c2$> // all pair shortest path or the *pairwise-dist()* task as defined.

Constraints

$1 \leq N \leq 1000$.

$0 \leq L \leq N^2$.

$1 \leq C \leq 256$.

$0.0 \leq t(u, v) \leq 1000.0$.

$1 \leq k \leq N$.

Output Format

Depending on the task, follow the following output format:

min-time():

mt_1 mt_2 ... mt_N // min time from s to other articles. $mt_s = 0$. For unreachable articles, -1.

update():

Nothing to print.

top-k():

top_1 top_2 ... top_k // top-k articles referred by articles of category c.

pairwise-dist():

For every pair (u, v) of articles, print:

u v <tab> $shortest - path$ from u to v // if v is reachable from u.

u v <tab> -1 // if v is not reachable from u.

Note: All the floating-point numbers need to be printed till **exactly 3 decimal places**.

Sample Input 0

```
6 10
1 1 2 1 2 3
1 2 10.0
1 3 1.0
2 1 5.0
2 3 1.0
2 5 5.0
4 1 4.0
4 3 3.0
4 5 2.0
5 3 1.0
6 5 1.0
sssp 1
apsp 1 2
update 1 7.0
apsp 1 3
top-k 1 3
```

Sample Output 0

```
0.000 10.000 1.000 -1.000 15.000 -1.000
1 3    1 3
1 5    1 2 5
2 3    2 3
2 5    2 5
4 3    4 3
4 5    4 5
1 6    1 6
2 6    2 1 6
4 6    4 1 6
3 5 1
```

Explanation 0

Read the explanation from PDF on moodle.

Sample Input 1

```
10 36
4 2 2 2 1 2 4 3 4 1
7 3 23.868
6 9 7.06
1 3 27.014
4 8 93.665
5 4 8.651
8 9 81.284
6 2 83.149
3 7 26.716
2 5 11.638
5 8 88.041
1 2 48.25
10 8 87.116
6 7 91.24
3 10 34.193
6 3 11.474
1 5 89.253
3 5 39.777
10 9 51.852
3 2 97.318
2 6 82.907
10 4 58.189
7 1 41.779
9 3 21.029
7 5 43.186
4 2 86.269
6 5 12.818
7 9 20.3
2 7 74.222
8 3 52.691
6 8 73.812
7 8 2.976
5 7 23.544
1 8 10.158
4 3 5.652
5 2 60.202
10 2 72.463
sssp 9
top-k 3 8
apsp 3 4
top-k 4 2
sssp 1
sssp 9
update 2 74.599
apsp 2 4
apsp 3 1
top-k 2 7
```

Sample Output 1

```
89.524 118.347 21.029 69.457 60.806 201.254 47.745 50.721 0.000 55.222
3 9
8 1    8 3 7 1
8 7    8 3 7
```

```
8 9   8 9
3 5
0.000 48.250 27.014 68.539 59.888 131.157 53.730 10.158 74.030 61.207
89.524 118.347 21.029 69.457 60.806 201.254 47.745 50.721 0.000 55.222
2 1   2 1
2 7   2 5 7
2 9   2 5 7 9
3 1   3 7 1
3 7   3 7
3 9   3 7 9
4 1   4 3 7 1
4 7   4 3 7
4 9   4 3 7 9
6 1   6 5 7 1
6 7   6 5 7
6 9   6 9
8 5   8 3 5
8 10   8 3 10
2 5 7 3 8 1 6
```