CYBERLAND SOLUTION

STATEMENT

Given a undirected weighted graph with N nodes and M edges. Some nodes can clear the current distance to 0, while some can divide the distance by 2. In each visit, you can only use the special ability at most once, and you can choose not to use the ability on the node.

Find the shortest path from 0 to H, if you can not visit H twice, and the divide-by-2 ability can be used at most K times.

N, M <= 10^5, K <= 10^6

STATEMENT?

Given a undirected weighted graph with N nodes and M edges. Some nodes can clear the current distance to 0, while some can divide the distance by 2. In each visit, you can only use the special ability at most once, and **you must use the ability** on the node.

Find the shortest path from 0 to H, if you can not visit H twice, and the divide-by-2 ability can be used at most K times.

N, M <= 10^5, K <= 30

SUBTASK 1

n<=3

Either go directly H, or go to H after passing through the third node.

Because 0, H has no special abilities, it is obviously not beneficial to repeatedly visit the third point.

Time complexity O(1)

SUBTASK 2,5

No Special ability。

A simple single source shortest path(SSSP) problem which can be solved through Dijkstra algorithm.

Time Complexity O(M log n)。

SUBTASK 3,6

No divide-by-2 ability.

Find the set of node S, which have set-zero ability, and can be reached from 0 without go through H.

If we reach any node in S, we can clear the distance to 0. So it’s a multiple source shortest path problem with {S} U{0} as the sourse set.

Use Dijkstra again and the time complexity is also O(M log N).

SUBTASK 4,7

K <= 30

Same as the idea of subtask 3,6, we can find the set S, and treat the problem as a multi-source shortest path problem.

Since K <= 30, we can use dynamic programming, and the state go as follows:

f[i][t]: The shortest distance to node i, if exactly t time of special ability is used.

To transfer between states, we need to consider whether to use the special ability.

- If we do not use special ability, we can treat transition as another multi-source shortest path problem.
- If we use special ability, t is increased by 1, we can transfer in order of increasing t.

Time complexity O(MK log N).

SUBTASK 8

K <= 1000000。

Suspect that K is not required to be very large.

Accepted.

WHY?

We proof that for every possible V, assume that we can use at most V times of divide-by-2 ability, we can always make the absolute errors at most N max(C) / 2^K, If we use the algorithm for subtask 7.

Here, max(C) represents the maximum cost to pass through an edge。

WHY?

1. Assume that the optimal solution for at most i-times divide-by-2 ability usage is ans[i]. Then ans[i] is non-increasing.

2. Find the optimal solution of at most V-times divide-by-2 ability usage(V >> K holds). We call this the original solution.

3. Keep the last K divide-by-two ability usage of the original solution , and delete all previous divide-by-two ability usage. We call this the intermediate solution.

4. Assuming that the intermediate path uses divided-by-two ability first in node p. Replace the path before first divided-by-two ability usage, with the shortest path from 0 to p without using divide-by-2 ability. We call this path the edited solution.

5. The shortest path from 0 to p is at most N max{C}, and edited solution preserves the path of the original solution after the last K divide-by-two ability usage. So it is at most N max{C} / 2^K worse than the original solution.

NOTE

Take $K = \min\{K, 70\}$, and an absolute error of at most 1e-7 can be achieved.

Time complexity $O(m \log n (\log n + \log \max\{V\} - \log \text{epsilon}))$.