Problem Menu Back

Statement

Submissions

Your score not attempted

Spoilers

Show difficulty Show tags

solved by 29 / 30

graph: shortest path

Top users by time

#	User	Time
1	levirs565	129 ms
2	irdacin	129 ms
3	AhmadRomy	131 ms
4	JUstRadr	133 ms
5	Arman10	147 ms

Top users by memory

#	User	Memory
1	zhoelzx	8152 KB
2	levirs565	9032 KB
3	vandykas	9196 KB
4	AhmadRomy	9416 KB
5	Arman10	10568 KB

Submission #3542045

Arkavidia 9.0 - Penyisihan CP / G. Gerbang Sihir Arkavidia

Accepted • levirs565 • C++20 • April 25, 2025 at 22:33:30

▶ Sample Test Data Results

▶ Test Data Results

```cpp
#include <bits/stdc++.h>

using namespace std;

#define INF 1e18
struct Edge
{
    int to, travelTime, switchTime;
    int firstState;
};

struct State
{
    long long currTime;
    int node;
    bool operator>(const State &other) const
    {
        return currTime > other.currTime;
    }
};
long long magicBridge(int N, int K, vector<vector<Edge>> &graph, int X, int Y)
{
    vector<long long> dist(N + 1, INF);
    priority_queue<State, vector<State>, greater<State>> pq;
    pq.push({0, X});
    dist[X] = 0;

    while (!pq.empty())
    {
        State top = pq.top();
        pq.pop();
        long long currTime = top.currTime;
        int node = top.node;

        if (node == Y)
            return currTime;
        if (currTime > dist[node])
            continue;

        for (auto &edge : graph[node])
        {
            int neighbor = edge.to;
            bool dapatDilewati = (currTime / edge.switchTime + edge.firstState) % 2 == 0;
            // cout << currTime << " " << edge.switchTime <<  " " << edge.firstState << endl;
            long long nextTime = currTime + edge.travelTime;
            if (!dapatDilewati) {
                // cout << "Tidak dapat" << endl;
                nextTime += edge.switchTime - currTime % edge.switchTime;
            }
            if (nextTime < dist[neighbor])
            {
                dist[neighbor] = nextTime;
                pq.push({nextTime, neighbor});
            }
        }
    }
    return -1;
}
void solution()
{
    int n, k;
    cin >> n >> k;
    vector<vector<Edge>> graf(n + 1);
    for (int i = 0; i < k; i++)
    {
        int a, b, c, t;
        cin >> a >> b >> c >> t;
        graf[a].push_back({b, c, t, 0});
        graf[b].push_back({a, c, t, 1});
    }
    int x, y;
    cin >> x >> y;
    cout << magicBridge(n, k, graf, x, y) << endl;
}
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(0);
    int t;
    t = 1;
    // cin >> t;
    while (t--)
    {
        solution();
    }
    return 0;
}
```