

apsidorova@edu....

HSE_Graph_algorithms(2023)

4 июл 2023, 18:17:40

старт: 19 фев 2023, 10:00:00

финиш: 20 апр 2023, 10:00:00

длительность: 2м. 1д.

начало: 19 фев 2023, 10:00:00

конец: 20 апр 2023, 10:00:00

D. Одинокий верблюд

Ограничение времени	3 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Верблюд Петр очень не любит других верблюдов. Не любит настолько, что старается как можно реже встречаться с ними. К счастью для него пустыня очень большая, и поэтому такие встречи случаются достаточно редко. Вся жизнь Петра состоит в том, что он кочует от одного оазиса к другому, как и все другие верблюды. Между многими оазисами давным давно уже протоптаны пути, и все верблюды строго следуют им. Стоит заметить, что если есть путь из оазиса a в оазис b , то это не означает, что есть таковой из b в a (иначе на обратном пути иногда пришлось бы идти в гору, чего верблюды категорически не любят). Некоторые пути ведут из оазиса в него же. Между некоторыми оазисами может быть несколько путей.

Увы, но в момент перехода от одного оазиса к другому Петр может встретиться с несколькими верблюдами. Сейчас он находится у оазиса x и хочет попасть в оазис y . За долгие годы скитаний он составил карту путей, в которой для каждого пути также указано, сколько верблюдов повстречает Петр, если будет идти по этому пути. Помогите Петру добраться от одного оазиса к другому, встретив как можно меньше других верблюдов.

Формат ввода

В первой строке записано 2 целых числа через пробел — N ($1 \leq N \leq 5000$) и M ($0 \leq M \leq 20000$) — число оазисов в пустыне и число путей соответственно. Далее следует M строк, каждая из которых содержит 3 целых числа, разделенных пробелами — a_i , b_i ($1 \leq a_i, b_i \leq N$) и c_i ($0 \leq c_i \leq 2$), означающих, что на пути из a_i в b_i Петр повстречает c_i верблюдов. Далее следует целое число K ($0 \leq K \leq 10000$ и $0 \leq K \cdot N \leq 2 \cdot 10^7$) — количество запросов Петра. Наконец, каждая из следующих K строк содержит по 2 целых числа, разделенных пробелом — x_i и y_i ($1 \leq x_i, y_i \leq N$), означающих что в i -ом запросе Петру нужно добраться из x_i в y_i .

Формат вывода

Выведите ответ на каждый запрос в отдельной строке. Если в каком-либо запросе искомого пути не существует, выведите в качестве ответа на этот запрос число -1.

Пример

Ввод Вывод

5 4
1 2 2
2 3 0
2 4 2

3
-1

3 4 1
2
1 4
2 5

```
1 #include <iostream>
2 #include <vector>
3 #include <utility>
4 #include <queue>
5
6 int find_way(const int &in_vertex, const int &out_vertex,
7             const std::vector<std::vector<std::pair<int, int>>> &vertexes, const int &num_vertex) {
8     int weights = 3;
9     std::queue<int> the_way[weights];
10    std::vector<bool> used(num_vertex, false);
11    std::vector<int> distance(num_vertex, 20000);
12
13    the_way[0].push(in_vertex);
14    int index = 0;
15    used[in_vertex] = true;
16    distance[in_vertex] = 0;
17
18    while (!(the_way[0].empty() & the_way[1].empty() & the_way[2].empty())) {
19        while (!the_way[index].empty()) {
20            int now = the_way[index].front();
21            the_way[index].pop();
22
23            if (now == out_vertex) {
24                return distance[out_vertex];
25            }
26
27            used[now] = true;
28
29            for (auto &to: vertexes[now]) {
30                if (!used[to.first]) {
31                    if (distance[to.first] > distance[now] + to.second) {
32                        distance[to.first] = distance[now] + to.second;
33                        the_way[(to.second + index) % 3].push(to.first);
34                    }
35                }
36            }
37        }
38        index = (index + 1) % 3;
```