

Чемпионат по программированию: Бэкенд-разработка - Квалификация

3 ноя 2019, 17:50:58

старт: 2 ноя 2019, 20:02:59

финиш: 3 ноя 2019, 01:02:59

длительность: 05:00:00

начало: 14 окт 2019, 12:00:00

конец: 20 окт 2019, 23:59:59

А. Динамическое ценообразование

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В некотором городе есть N водителей и M пассажиров. В этом городе действует система динамического ценообразования: чем больше водителей находятся рядом с пассажиром, тем меньше будет стоить поездка. Положим, что базовая цена поездки для каждого пассажира равна K . Дано число P . Каждый водитель, находящийся на расстоянии I от некоторого пассажира (при $I < P$) уменьшает стоимость поездки на $P - I$, при этом цена не может быть меньше 1.

Назовём цену поездки с учётом снижения за счёт водителей, находящихся рядом, динамической ценой.

Найдите пассажира, для которого динамическая цена будет минимальной среди всех пассажиров.

Формат ввода

В первой строке даны целые числа M , N , P и K .

Во второй строке дано число S . В следующих S строках даны расстояния между водителями и пассажирами - три числа p m g , где p - номер водителя, m - номер пассажира, g - длина пути от водителя с номером p до пассажира с номером m - целые числа. Гарантируется, что $0 \leq p < N$, $0 \leq m < M$, $0 < g \leq 10000$.

Затем дано число T . В следующих T строках даны расстояния между пассажирами - три числа $m1$ $m2$ g , где $m1$ и $m2$ - номера пассажиров, а g - длина пути от пассажира с номером $m1$ до пассажира $m2$ - целые числа. Гарантируется, что $0 \leq m1, m2 < M$, $0 < g \leq 10000$.

Гарантируется, что $M, N \leq 2000$; $P \leq 100$; $K \leq 10000$; $S, T \leq 50000$

Формат вывода

Следует вывести одно число - минимальную среди всех пассажиров динамическую цену

Пример

Ввод**Вывод**

Ввод	Вывод
2 2 5 10 2 0 0 2 1 1 2 1 0 1 2	6

Примечания

Поясним пример: рассматриваются два пассажира (с номерами 0 и 1) и два водителя (с номерами 0 и 1). Существует путь от водителя 0 к пассажиру 0 длины 2 и от водителя 1 к пассажиру 1 длины 2. Кроме того, существует путь от пассажира 0 к пассажиру 1 длины 2. Таким образом, водитель 0 уменьшает цену для пассажира 0 на 3 ($= 5 - 2$), а для пассажира 1 - на 1 ($= 5 - (2 + 2)$). Водитель 1 уменьшает цену для пассажира 1 на 3, и не уменьшает цену пассажира 0, поскольку не существует пути от водителя 1 к пассажиру 0. Итого, имеем для пассажира 0 цену 7 ($= 10 - 3$), а для пассажира 1 - цену 6 ($= 10 - 3 - 1$).

Следует обратить внимание, что если указана длина пути от некоторой точки А до некоторой точки В, то это не значит, что путь из В в А имеет такую же длину или вообще существует. Если расстояние между какими-либо двумя точками не указано, это значит, что прямого пути между ними нет, но может быть путь, проходящий через одну или несколько промежуточных точек.

Язык GNU c++17 7.3

Набрать здесь

Отправить файл

```
1 #include <algorithm>
2
3 #include <queue>
4 #include <unordered_map>
5 #include <vector>
6
7 #include <iostream>
8
9 uint16_t M, N, P, K;
10
11 using pass_t = uint16_t;
12 using dist_t = uint16_t;
13 using price_t = uint16_t;
14
15 std::vector<std::unordered_map<pass_t, dist_t>> passengerEdges, driverEdges;
16 std::vector<price_t> prices;
17
18 void ReadInput();
19 void InspectDriver(const std::unordered_map<pass_t, dist_t>& distToPassengers);
20 void RunDijkstra(const std::unordered_map<pass_t, dist_t>& initialDistances, std::vector<dist_t>& distances);
21
22 int main() {
23     ReadInput();
24
25     for (const auto& driver : driverEdges)
26         InspectDriver(driver);
27
28     std::cout << *std::min_element(prices.cbegin(), prices.cend()) << std::endl;
29     return 0;
30 }
31
32 void CreateEdge(std::unordered_map<pass_t, dist_t>& edges, pass_t to, dist_t value) {
33     if (value >= P)
34         return;
35
36     auto existing = edges.find(to);
37     if (existing == edges.end() || existing->second > value)
38         edges[to] = value;
```

Отправить

Следующая