

Отчет

по лабораторной работе «1650. Миллиардеры»
по дисциплине «Алгоритмы и Структуры данных»

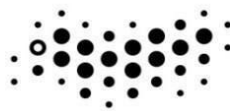
Авторы:

Полит Алексей Денисович

Факультет: СУиР

Группа: R3235

Преподаватель: Тропченко Андрей Александрович



УНИВЕРСИТЕТ ИТМО

1. Задача

Возможно, вы знаете, что из всех городов мира больше всего миллиардеров живёт в Москве. Но, поскольку работа миллиардера подразумевает частые перемещения по всему свету, в определённые дни какой-то другой город может занимать первую строчку в таком рейтинге. Ваши приятели из ФСБ, ФБР, MI5 и Шин Бет скинули вам списки перемещений всех миллиардеров за последнее время. Ваш работодатель просит посчитать, сколько дней в течение этого периода каждый из городов мира был первым по общей сумме денег миллиардеров, находящихся в нём.

2. Исходные данные

В первой строке записано число n — количество миллиардеров ($1 \leq n \leq 10000$). Каждая из следующих n строк содержит данные на определённого человека: его имя, название города, где он находился в первый день данного периода, и размер состояния. В следующей строке записаны два числа: m — количество дней, о которых есть данные ($1 \leq m \leq 50000$), k — количество зарегистрированных перемещений миллиардеров ($0 \leq k \leq 50000$). Следующие k строк содержат список перемещений в формате: номер дня (от 1 до $m - 1$), имя человека, название города назначения. Вы можете считать, что миллиардеры путешествуют не чаще одного раза в день, и что они отбывают поздно вечером и прибывают в город назначения рано утром следующего дня. Список упорядочен по возрастанию номера дня. Все имена и названия городов состоят не более чем из 20 латинских букв, регистр букв имеет значение. Состояния миллиардеров лежат в пределах от 1 до 100 миллиардов.

3. Текст программы

```
#include <map>
#include <iostream>
#include <string>
#include <set>
#include <unordered_map>

using namespace std;

int main() {
    unordered_map<string, long long> cityAndMoney;      // Город
    и его капитализированные деньги
    unordered_map<string, long long> richmanAndMoney;  // Богач
    и его сумма денег
```

```

        unordered_map<string, string> richmanAndCity;           // Богач
и его местоположение
        map<long long, set<string>> moneyAndCitiesSet;         //
Деньги и города, с такой суммой
        map<string, int> citiesRank;                           //
Финальный топ городов
        long long money;
        string name, city;
        int n;
        cin >> n;
        for (int i = 0; i < n; i++) {
            cin >> name >> city >> money;
            richmanAndMoney[name] = money;    //заполнение карты
связью {Богач-Деньги}
            richmanAndCity[name] = city;    //заполнение карты
связью {Богач-Город}
            if (cityAndMoney.find(city) != cityAndMoney.end()) {
                /*
                Если город city существует, значит над ним мы
                проводим операцию не первый раз, значит его
                капитализация изменится. Отчистим в карте
                {Капитализация-Города} связь (её больше не будет существовать).
                */
                long long sum = cityAndMoney[city];
                moneyAndCitiesSet[sum].erase(city);
                //Если города с такой капитализацией
                перестали существовать, отчистим и ячейку с множеством
                городов.
                if (moneyAndCitiesSet[sum].size() == 0)
                moneyAndCitiesSet.erase(sum);
            }
            cityAndMoney[city] += money; //увеличение
капитализации города
            moneyAndCitiesSet[cityAndMoney[city]].insert(city);
//заношим город в множество городов с такой капитализацией
        }
        int days, movements, day, prevDay, currDay = 0;
        cin >> days >> movements;
        for (int i = 0; i <= movements; i++) { //вычислим капитализации
        для каждого из городов в разные дни
            prevDay = currDay;
            //Обрабатываем ситуацию последнего дня, когда ещё
            нужно менять параметры сущностей, но данные вводить уже не
            надо.
            if (i == movements) day = days;
            else cin >> day >> name >> city;
            currDay = day;
            map<long long, set<string>>::reverse_iterator it =
moneyAndCitiesSet.rbegin();

```

```

        /*
        Если в отсортированном по ключам контейнере
        {Деньги-Список городов} в ячейке, соответствующей
        наибольшему ключу
        только один элемент в {set<string>}, значит сейчас
        существует единственный город с максимальной суммой, и ему
        нужно добавить дни в карту {Город-Количество дней в
        money}.
        */
        if (currDay != prevDay && it->second.size() == 1)
        citiesRank["*(it->second.begin())"] += currDay - prevDay;
        if (i < movements) {
            /*
            Старое местоположение богача: если он уехал из
            города, необходимо поменять соответствующую
            информацию во всех картах.
            */
            string oldLocation = richmanAndCity[name];
            long long oldMoney = cityAndMoney[oldLocation];
            moneyAndCitiesSet[oldMoney].erase(oldLocation);
            if (moneyAndCitiesSet[oldMoney].size() == 0)
            moneyAndCitiesSet.erase(oldMoney);
            cityAndMoney[oldLocation] -=
            richmanAndMoney[name];

            moneyAndCitiesSet[cityAndMoney[oldLocation]].insert(oldLocation);
            /*
            Новое местоположение богача: если он приехал в
            новый город, необходимо поменять соответствующую
            информацию во всех картах.
            */
            long long newMoney = cityAndMoney[city];
            moneyAndCitiesSet[newMoney].erase(city);
            if (moneyAndCitiesSet[newMoney].size() == 0)
            moneyAndCitiesSet.erase(newMoney);
            cityAndMoney[city] += richmanAndMoney[name];
            moneyAndCitiesSet[cityAndMoney[city]].insert(city);
            richmanAndCity[name] = city;
        }
    }
    for (const auto& c: citiesRank) cout << c.first << " " << c.second <<
    endl; //вывод мона городов
    return 0;
}

```

9259635	16:44:40 9 мар 2021	Aleksey	1650. Миллиардеры	G++ 9.2 x64	Accepted		0.671	12 840 КБ
---------	------------------------	-------------------------	-----------------------------------	-------------	----------	--	-------	-----------

4. Описание программы

Реализация данного алгоритма не очень сложная. Нужно просто аккуратно работать с данными и поддерживать связь. Связи должны поддерживаться между всеми сущностями. Для них подходят `{map}` и `{unordered_map}`. Одно ключ, другое - значение. Такие карты должны быть для всех нужных связей