ВШЭ АиСД 2021. Краскал, фильтр Блума

6 дек 2021, 16:20:56
старт: 28 ноя 2021, 16:00:00
финиш: 8 дек 2021, 23:59:00
до финиша: 2д. 7ч.
начало: 28 ноя 2021, 16:00:00
конец: 8 дек 2021, 23:59:00
длительность: 10д. 7ч.

A. Минимальный каркас (0.25)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

От вас требуется определить вес минимального остовного дерева для неориентированного взвешенного связного графа.

Формат ввода

В первой строке входных данных находятся числа N и M (2 <= N <= 20000; 1 <= M <= 100000), где N – количество вершин в графе, а M – количество рёбер. В каждой из последующих M строк записано по тройке чисел A, B, C, где A и B – номера вершин, соединённых ребром, а C – вес ребра (натуральное число, не превышающее 1000000).

Формат вывода

Вывести одно число – искомый вес.

Пример

Ввод	Вывод
3 3 1 2 1 2 3 2 3 1 3	3

Примечания

Система оценки

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		N	M		
0	1	–	–	–	Тест из условия.
1	2	$N \le 10$	$M \le 100$	0	
2	2	$N \le 100$	$M \le 6000$	0 – 1	
3	3	$N \le 10000$	$M \le 100000$	0 – 2	
4	2	$N \le 20000$	$M \le 100000$	0 – 3	**Offline-проверка**

Набрать здесь | Отправить файл

```cpp
#include <iostream>

struct Edge {
    Edge() {
    }

    int firstVertex;
    int secondVertex;
    int distance;
};

int find(int index, int *parent) {
    if (index == parent[index]) {
        return index;
    }

    return parent[index] = find(parent[index], parent);
}

bool unions(int first_node, int second_node, int *parent) {
    first_node = find(first_node, parent);
    second_node = find(second_node, parent);

    if (first_node == second_node) {
        return false;
    }

    parent[second_node] = first_node;
    return true;
}

void mySwap(Edge *first, Edge *second) {
    Edge temp = *first;
    *first = *second;
    *second = temp;
}

int partition(Edge *array, int low, int high) {
```

Отправить

Следующая