

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**ОТЧЕТ по практической работе**

**по дисциплине “Методы программирования”**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автор |  | | |  | Люсюков А.В. | | | |
|  | (подпись, дата) | | |  |  | | | |
| Обозначение | 10.05.01.550000.000 О | | Группа | | | | ВКБ31 |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | | | | |
| Профиль | Компьютерная безопасность | | | | | | |
| Преподаватель |  | | |  | | Савельев В.А. | |
|  | (подпись, дата) | | |  | |  | |

г. Ростов-на-Дону

2021 год

**Лабораторная работа №5**

Требуется найти в связном графе остовное дерево минимально веса.

**Входные данные**

Первая строка входного файла содержит два натуральных числа n и m - количество вершин и ребер графа соответственно (1≤n≤20000, 0≤m≤100000). Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами bi, ei и wi - номера концов ребра и его вес соответственно (1≤bi,ei≤n, 0≤wi≤100000).

Граф является связным.

**Выходные данные**

Выведите единственное целое число - вес минимального остовного дерева.

**Примеры**

|  |  |
| --- | --- |
| входные данные  4 4  1 2 1  2 3 2  3 4 5  4 1 4 | выходные данные  7 |

#include<iostream>

#include<vector>

#include<set>

#include<algorithm>

#include<cstdlib>

#define MP make\_pair

using namespace std;

vector<int> p;

int find\_set(int x){return(x==p[x])?x:p[x]=find\_set(p[x]);}

void unite\_set(int x, int y){

x=find\_set(x);y=find\_set(y);if(x==y)return;

if(rand()%2)swap(x,y);

p[x]=y;}

int main(){

int n,m,i,j,k,s=0,w;long long ans=0;

cin>>n>>m;

p.resize(n+1);for(k=1;k<=n;++k)p[k]=k;

pair<int,pair<int, int>>r[m];

for(k=0;k<m;++k){cin>>i>>j>>w;r[k]=MP(w,MP(i,j));}

sort(r, r+m);

for(k=0;k<m&&s<n-1;++k){

i=find\_set(r[k].second.first);

j=find\_set(r[k].second.second);

if(i!=j){

ans+=r[k].first;s++;unite\_set(i,j);}}

cout<<ans;

return 0;}

