#include<bits/stdc++.h>

using namespace std;

vector<int> mst\_tree(vector<vector<pair<int,int>> > &graph){

int n = graph.size();

vector<int> par(n,-1);

vector<int> key(n,INT\_MAX);

vector<bool> mst(n,false);

key[0] = 0;

// pq = {wt,node}

priority\_queue<pair<int,int>, vector<pair<int,int> >, greater<pair<int,int> > > pq;

pq.push({0,0});

while(pq.size()>0){

int node = pq.top().second;

int wt = pq.top().first;

pq.pop();

mst[node] = true;

for(auto it: graph[node]){

if(mst[it.first]==false && key[it.first]>it.second){

key[it.first] = it.second;

par[it.first] = node;

pq.push({it.second,it.first});

}

}

}

int cost = 0;

for(int i=0;i<key.size();i++){

cost+= key[i];

}

cout<<cost<<endl;

return par;

}

int main(){

int n;

int e;

cin>>n>>e;

vector<vector<pair<int,int> > > graph(n);

while(e--){

int x,y,w;

cin>>x>>y>>w;

graph[x].push\_back({y,w});

graph[y].push\_back({x,w});

}

vector<int> ans = mst\_tree(graph);

// for(int i=1;i<n;i++){

// cout<<ans[i]<<"-> "<<i<<endl;

// }

}