



Signup and get free access to 100+ Tutorials and Practice Problems

[Start Now](#)[All Tracks](#) > [Algorithms](#) > [Graphs](#) > Minimum Spanning Tree

Algorithms

3
LIVE EVENTS

Solve any problem to achieve a rank

[View Leaderboard](#)Topics:

Minimum Spanning Tree

[TUTORIAL](#) [PROBLEMS](#)

Pre-requisites: Graphs, Trees

What is a Spanning Tree?

Given an undirected and connected graph $G = (V, E)$, a spanning tree of the graph G is a tree that spans G (that is, it includes every vertex of G) and is a subgraph of G (every edge in the tree belongs to G)

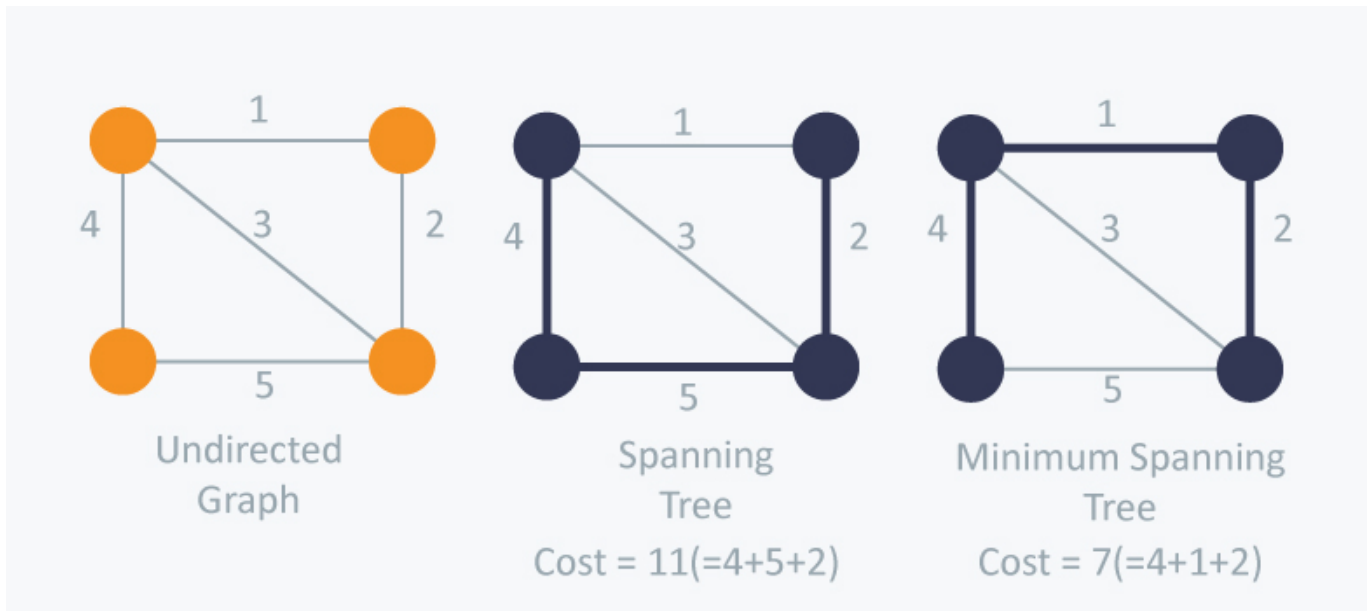
What is a Minimum Spanning Tree?

The cost of the spanning tree is the sum of the weights of all the edges in the tree. There can be many spanning trees. Minimum spanning tree is the spanning tree where the cost is minimum among all the spanning trees. There also can be many minimum spanning trees.

Minimum spanning tree has direct application in the design of networks. It is used in algorithms approximating the travelling salesman problem, multi-terminal minimum cut problem and minimum-cost weighted perfect matching. Other practical applications are:

1. Cluster Analysis
2. Handwriting recognition
3. Image segmentation

?



There are two famous algorithms for finding the Minimum Spanning Tree:

Kruskal's Algorithm

Kruskal's Algorithm builds the spanning tree by adding edges one by one into a growing spanning tree. Kruskal's algorithm follows greedy approach as in each iteration it finds an edge which has least weight and add it to the growing spanning tree.

Algorithm Steps:

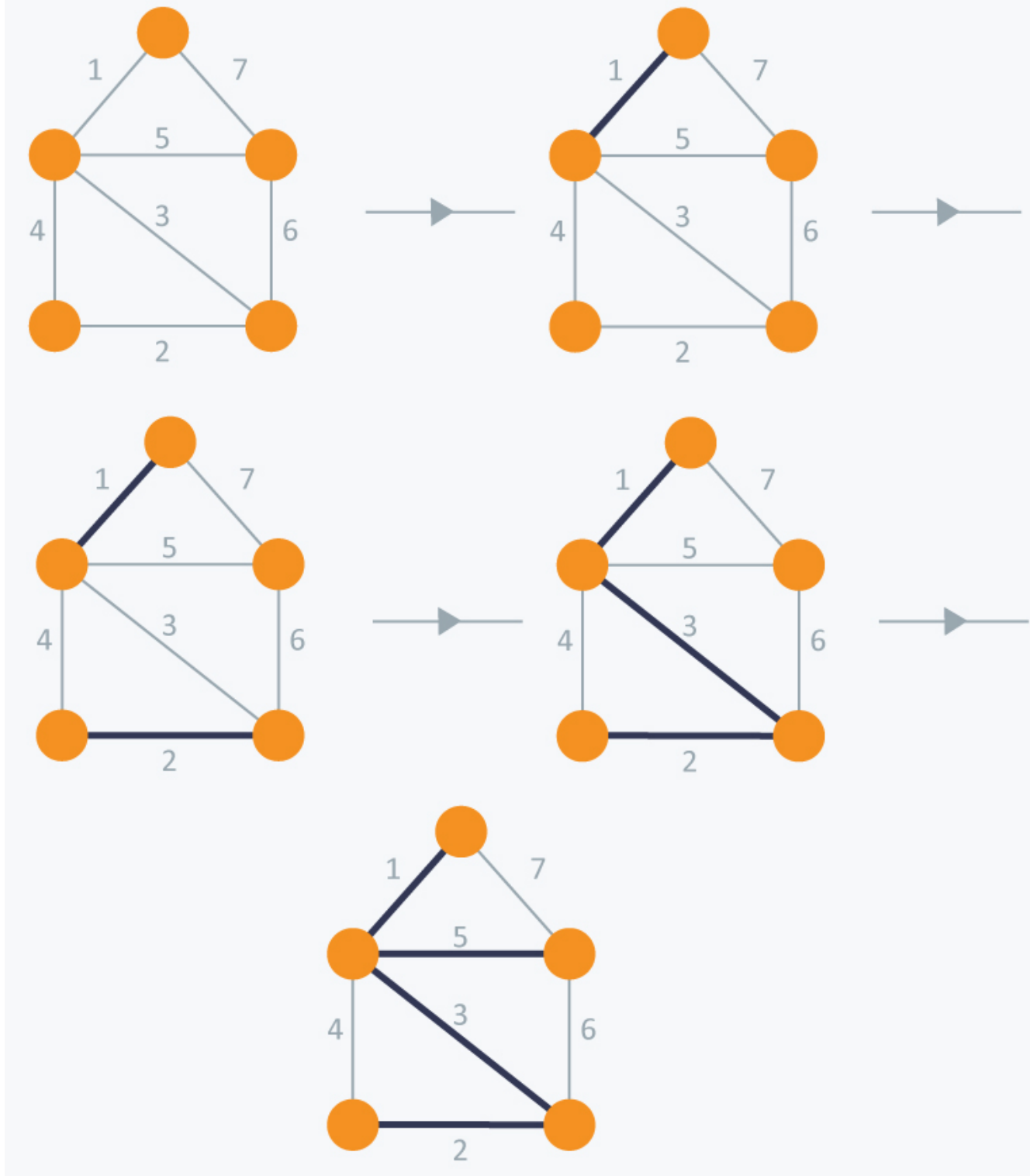
- Sort the graph edges with respect to their weights.
- Start adding edges to the MST from the edge with the smallest weight until the edge of the largest weight.
- Only add edges which doesn't form a cycle , edges which connect only disconnected components.

So now the question is how to check if **2** vertices are connected or not ?

This could be done using DFS which starts from the first vertex, then check if the second vertex is visited or not. But DFS will make time complexity large as it has an order of $O(V + E)$ where V is the number of vertices, E is the number of edges. So the best solution is "**Disjoint Sets**": Disjoint sets are sets whose intersection is the empty set so it means that they don't have any element in common.

Consider following example:

Kruskal's Algorithm



In Kruskal's algorithm, at each iteration we will select the edge with the lowest weight. So, we will start with the lowest weighted edge first i.e., the edges with weight 1. After that we will select the second lowest weighted edge i.e., edge with weight 2. Notice these two edges are totally disjoint. Now, the next edge will be the third lowest weighted edge i.e., edge with weight 3, which connects the two

disjoint pieces of the graph. Now, we are not allowed to pick the edge with weight 4, that will create a cycle and we can't have any cycles. So we will select the fifth lowest weighted edge i.e., edge with weight 5. Now the other two edges will create cycles so we will ignore them. In the end, we end up with a minimum spanning tree with total cost 11 ($= 1 + 2 + 3 + 5$).

Implementation:

```
#include <iostream>
#include <vector>
#include <utility>
#include <algorithm>

using namespace std;
const int MAX = 1e4 + 5;
int id[MAX], nodes, edges;
pair <long long, pair<int, int> > p[MAX];

void initialize()
{
    for(int i = 0; i < MAX; ++i)
        id[i] = i;
}

int root(int x)
{
    while(id[x] != x)
    {
        id[x] = id[id[x]];
        x = id[x];
    }
    return x;
}

void union1(int x, int y)
{
    int p = root(x);
    int q = root(y);
    id[p] = id[q];
}

long long kruskal(pair<long long, pair<int, int> > p[])
{
    int x, y;
    long long cost, minimumCost = 0;
```

?

```

    for(int i = 0; i < edges; ++i)
    {
        // Selecting edges one by one in increasing order from the
        // beginning
        x = p[i].second.first;
        y = p[i].second.second;
        cost = p[i].first;
        // Check if the selected edge is creating a cycle or not
        if(root(x) != root(y))
        {
            minimumCost += cost;
            union1(x, y);
        }
    }
    return minimumCost;
}

int main()
{
    int x, y;
    long long weight, cost, minimumCost;
    initialize();
    cin >> nodes >> edges;
    for(int i = 0; i < edges; ++i)
    {
        cin >> x >> y >> weight;
        p[i] = make_pair(weight, make_pair(x, y));
    }
    // Sort the edges in the ascending order
    sort(p, p + edges);
    minimumCost = kruskal(p);
    cout << minimumCost << endl;
    return 0;
}

```

Time Complexity:

In Kruskal's algorithm, most time consuming operation is sorting because the total complexity of the Disjoint-Set operations will be $O(E \log V)$, which is the overall Time Complexity of the algorithm.

Prim's Algorithm

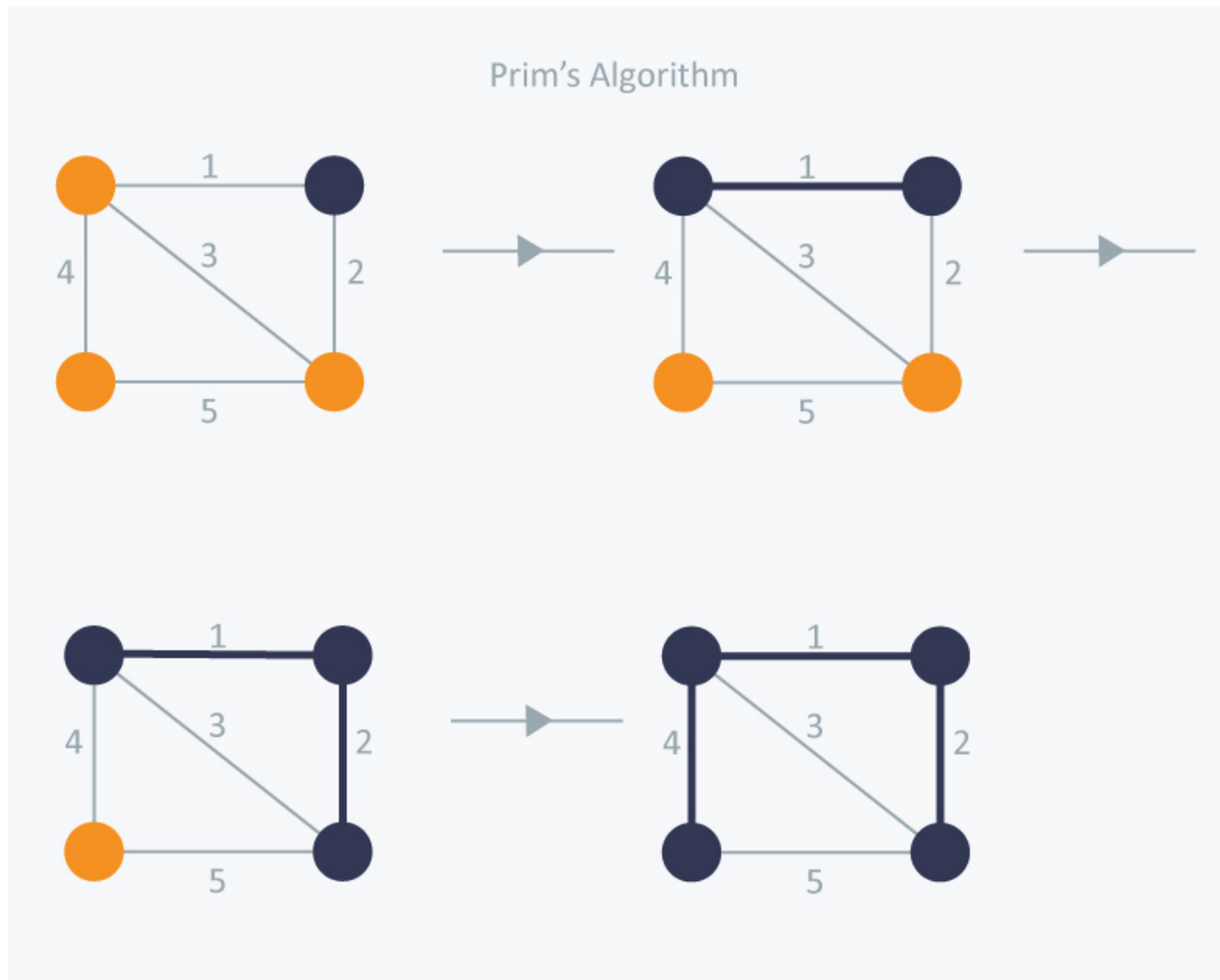
Prim's Algorithm also use Greedy approach to find the minimum spanning tree. In Prim's Algorithm we grow the spanning tree from a starting position. Unlike an **edge** in Kruskal's, we add **vertex** to the growing spanning tree in Prim's.

?

Algorithm Steps:

- Maintain two disjoint sets of vertices. One containing vertices that are in the growing spanning tree and other that are not in the growing spanning tree.
- Select the cheapest vertex that is connected to the growing spanning tree and is not in the growing spanning tree and add it into the growing spanning tree. This can be done using Priority Queues. Insert the vertices, that are connected to growing spanning tree, into the Priority Queue.
- Check for cycles. To do that, mark the nodes which have been already selected and insert only those nodes in the Priority Queue that are not marked.

Consider the example below:



In Prim's Algorithm, we will start with an arbitrary node (it doesn't matter which one) and mark it. In each iteration we will mark a new vertex that is adjacent to the one that we have already marked. As a greedy algorithm, Prim's algorithm will select the cheapest edge and mark the vertex. So we will simply choose the edge with weight 1. In the next iteration we have three options, edges with weight 2, 3 and 4. So, we will select the edge with weight 2 and mark the vertex. Now again we have three options, edges with weight 3, 4 and 5. But we can't choose edge with weight 3 as it is creating a cycle ?

So we will select the edge with weight 4 and we end up with the minimum spanning tree of total cost 7 ($= 1 + 2 + 4$).

Implementation:

```
#include <iostream>
#include <vector>
#include <queue>
#include <functional>
#include <utility>

using namespace std;
const int MAX = 1e4 + 5;
typedef pair<long long, int> PII;
bool marked[MAX];
vector <PII> adj[MAX];

long long prim(int x)
{
    priority_queue<PII, vector<PII>, greater<PII> > Q;
    int y;
    long long minimumCost = 0;
    PII p;
    Q.push(make_pair(0, x));
    while(!Q.empty())
    {
        // Select the edge with minimum weight
        p = Q.top();
        Q.pop();
        x = p.second;
        // Checking for cycle
        if(marked[x] == true)
            continue;
        minimumCost += p.first;
        marked[x] = true;
        for(int i = 0; i < adj[x].size(); ++i)
        {
            y = adj[x][i].second;
            if(marked[y] == false)
                Q.push(adj[x][i]);
        }
    }
    return minimumCost;
}
```

?

```

int main()
{
    int nodes, edges, x, y;
    long long weight, minimumCost;
    cin >> nodes >> edges;
    for(int i = 0; i < edges; ++i)
    {
        cin >> x >> y >> weight;
        adj[x].push_back(make_pair(weight, y));
        adj[y].push_back(make_pair(weight, x));
    }
    // Selecting 1 as the starting node
    minimumCost = prim(1);
    cout << minimumCost << endl;
    return 0;
}

```

Time Complexity:

The time complexity of the Prim's Algorithm is $O((V + E)\log V)$ because each vertex is inserted in the priority queue only once and insertion in priority queue take logarithmic time.

Contributed by: omar khaled abdelaziz abdelnabi

Did you find this tutorial helpful?



YES



NO

TEST YOUR UNDERSTANDING**Minimum Spanning Tree**

Given a weighted undirected graph. Find the sum of weights of edges of a Minimum Spanning Tree.

Input:

Given 2 integers N and M . N represents the number of vertices in the graph. M represents the number of edges between any 2 vertices.

Then M lines follow, each line has 3 space separated integers a_i , b_i , w_i where a_i and b_i represents an edge from a vertex a_i to a vertex b_i and w_i represents the weight of that edge.

Output:

Print the summation of edges weights in the MST.

?

Constraints:

$$2 \leq N \leq 10000$$

$$1 \leq M \leq 100000$$

$$1 \leq a_i, b_i \leq N$$

$$1 \leq w_i \leq 1000$$

SAMPLE INPUT

```
4 5
1 2 7
1 4 6
4 2 9
4 3 8
2 3 6
```

SAMPLE OUTPUT

```
19
```

Enter your code or [Upload your code](#) as file.[Save](#)

C (gcc 5.4.0)



```
1  /*
2  // Sample code to perform I/O:
3  #include <stdio.h>
4
5  int main(){
6      int num;
7      scanf("%d", &num);           // Reading input from STDIN
8      printf("Input number is %d.\n", num); // Writing output to STDOUT
9  }
10
11 // Warning: Printing unwanted or ill-formatted data to output will cause the test cases to fail
12 */
13
14 // Write your code here
15
```

1:1

☒ Provide custom input

COMPILE & TEST

SUBMIT



COMMENTS (87) SORT BY: **Relevance**▼

Login/Signup to Comment

**Gvs Akhil** 2 years ago

One of the best tutorials I've read till now. Most of the others are pretty outdated compared to this one.

Thanks!

▲ 68 votes ● Reply ● Message ● Permalink

**HIMADRI ANJOY** a year ago

yes man,its true

▲ 4 votes ● Reply ● Message ● Permalink

**Mrinal Verma**  Edited a year ago

priority_queue<Pll, vector<Pll>, greater<Pll> > Q;;can someone plz explain this line in the prims algorithm

▲ 3 votes ● Reply ● Message ● Permalink

**Mrinal Verma**  Edited 6 months ago

priority_queue<Pll, vector<Pll>, greater<Pll> > Q;

this statement means that the underlying container of the priority queue is vector<Pll>(which is by default is vector<int>), and greater<Pll> means that higher priority will be given to the smaller values.

In short it is a priority queue which will contain elements of form pair<long long ,int>, and which will act as a min-heap,therefore top-most element would be the minimum element.

for further details see the links below

<http://stackoverflow.com/questions/37227012/declaration-of-priority-queue-using-greater-and-vector>

http://www.cplusplus.com/reference/queue/priority_queue/

hope this would save someone's time

▲ 40 votes ● Reply ● Message ● Permalink

**Sachin Kumar** a year ago

thanks

▲ 0 votes ● Reply ● Message ● Permalink

**shivam kumar** a year ago

General definition of priority queue:

priority_queue<<type>,vector<type>,greater<type>>name;

1.type=data type to be inserted in queue;

2.vector<type>=it creates a vector of specified type("type") to store data of data type "type".

3.greaer<type>=it tells about priority ."greater"means it will give small integers greater priority and place it at top &larger values are inserted at end.

If greater is not written it means it will give large integers greater priority and place it at top &smaller values are inserted at end.

Ex-priority_queue<int>=means datatype inserted and stored in vector is int and priority is to take large numbers at top and smaller at the end.

▲ 10 votes ● Reply ● Message ● Permalink

**Panos Petridis** 2 years ago

Quick Note: If you use a Fibonacci Heap Prim's Algorithm can run in $O(E + (V \log(V)))$ time complexity which is significantly faster than Kruskal's algorithm when you have a network with a lot of edges. Which I think is worth mentioning. Other than that I think that this is a great tutorial.

▲ 9 votes ● Reply ● Message ● Permalink

?

**Ajay Verma** 10 months ago

infact there is problem.. one can try CHEFGAME on this concept on codechef :P

▲ 1 vote ● Reply ● Message ● Permalink

**Sharvin Jondhale** Edited a year ago

I think I've found an error in kruskal implementation of union find.. carefully see the root function ->

```
int root(int x)
```

```
{
  while(id[x] != x)
```

```
{
  id[x] = id[id[x]];
```

```
x = id[x]; / <- error is here, if x=id[x] then after first iteration the loop will stop, ALWAYS, which is wrong
```

```
}
return x;
}
```

instead I think we should have ->

```
...
while(id[x]!=x)
```

```
{
  x=id[x];
  id[x]=id[id[x]];
}
```

...
EDIT: The Original Algorithm is correct. There is no error.

▲ 0 votes ● Reply ● Message ● Permalink

**Priyanshu Varshney** a year ago

Visit this link and see union with path compression

<https://www.hackerearth.com/practice/data-structures/disjoint-data-strutures/basics-of-disjoint-data-structures/tutorial/>

▲ 5 votes ● Reply ● Message ● Permalink

**Sharvin Jondhale** a year ago

I did read this page before writing the comment, but I still think the while loop will stop after 1 iteration

▲ 0 votes ● Reply ● Message ● Permalink

**Ish Kool** a year ago

thanks i didn't have knowledge of disjoint sets before reading kruskal's, so was finding difficulty in understanding the implementation your comment saved the day!

▲ 1 vote ● Reply ● Message ● Permalink

**Manveer Singh** 7 months ago

Hackerearth you are contributing amazing articles to the community. Thanks!

▲ 2 votes ● Reply ● Message ● Permalink

**Anindya Pal** Edited a year ago

Even I thought so. But think about it this way..

say $i = 3$ initially when entering the function and suppose $id[3] = 4$, $id[4] = 5$, $id[5] = 6$.After an iteration, $id[i] = id[id[i]]$ is equivalent to $id[3] = id[4]$ which is equal to 5, now $i = 5$.But in condition of while loop we are comparing i and $id[new\ i = 5]$, i.e $i = 5$ and $id[5] = 6$.

This leads to next iteration.

Hope this helps :)

▲ 1 vote ● Reply ● Message ● Permalink

**Sharvin Jondhale** a year ago

That is correct ! Thank you very much.

?

▲ 1 vote ● Reply ● Message ● Permalink



Arijit Das 4 months ago

This is not a problem!!

▲ 0 votes ● Reply ● Message ● Permalink



Mouhssine Assaoud 2 years ago

I don't understand why the first algorithm gives a wrong answer when I submitted it . can someone explain to me ?

▲ 0 votes ● Reply ● Message ● Permalink



Ashank Anshuman 2 years ago

See the constraints : $1 \leq M \leq 100000$

While in the implementation of the first algorithm MAX is set as $10e4 + 5$.

▲ 5 votes ● Reply ● Message ● Permalink



samarth gupta a year ago

It has nothing to do with MAX limit. $MAX = 1e4 + 5$ is fine but still no idea why first algorithm is not giving correct answer.

▲ 0 votes ● Reply ● Message ● Permalink



Yash Aggarwal a year ago

No you have to change the MAX then it will give the correct answer

▲ 3 votes ● Reply ● Message ● Permalink



Angom Geetchandra 4 months ago

It does. Because MAX is used for number of vertices as well as EDGES! (pair <long long, pair<int, int> > p[MAX];) . And the input has 100000 edges, larger than MAX. Hence the wrong answer.

▲ 0 votes ● Reply ● Message ● Permalink



shivam kumar a year ago

General definition of priority queue:

`priority_queue<<type>,vector<type>,greater<type>>>name;`

1.type=data type to be inserted in queue;

2.vector<type>=it creates a vector of specified type("type") to store data of data type "type".

3.greaser<type>=it tells about priority ."greater"means it will give small integers greater priority and place it at top &larger values are inserted at end.

If greater is not written it means it will give large integers greater priority and place it at top &smaller values are inserted at end.

Ex-priority_queue<int>=means datatype inserted and stored in vector is int and priority is to take large numbers at top and smaller at the end.

▲ 2 votes ● Reply ● Message ● Permalink



Nikhil Joy Edited a year ago

While using unweighted graphs, i used array of lists(Adjacency list) to store connections, as I don't have to store the weight of the edge.

But now i am stuck here . Which Data structure should i use for weighted graph to store both connections and weight?(In Java)

Do i need to construct my own class to store these details?

e.g class edge

```
{
int x;
int y;
int weight;
}
```

?

▲ 1 vote ● Reply ● Message ● Permalink



Sachin Kumar ✎ Edited a year ago

Accepted Ans:-----

```
#include <bits/stdc++.h>
using namespace std;
const int MAX = 1e5+5;
typedef pair<long ,int > PII;
bool marked[MAX];
vector <PII> adj[MAX];

long prim(int x){
    priority_queue<PII , vector<PII> , greater<PII> > Q;
    int y;
    long long minimumCost = 0;
    PII p;
    Q.push(make_pair(0,x));
    while(!Q.empty()){
        p = Q.top();
        Q.pop();
        x = p.second;
        if(marked[x] == true){
            continue;
        }
        minimumCost += p.first;
        marked[x] = true;
        for(int i=0;i<adj[x].size();i++){
            y = adj[x][i].second;
            if(marked[y] == false)
                Q.push(adj[x][i]);
        }
    }
    return minimumCost ;
}

int main(void){
    int x,y,nodes,edges;
    long w , cost ,minimumCost;
    cin>>nodes>>edges;
    for(int i=0;i<edges;i++){
        cin>>x>>y>>w;
        adj[x].push_back(make_pair(w,y));
        adj[y].push_back(make_pair(w,x));
    }
    minimumCost = prim(1);
    cout<<minimumCost<<endl;
    return 0;
}
```

▲ 1 vote ● Reply ● Message ● Permalink



Vibhor Shukla 10 months ago

why do we need to check for this condition

if(marked[x] == true)

continue;

as the algorithm only sends those edges in the queue which are unmarked..

▲ 0 votes ● Reply ● Message ● Permalink



Vedant Gawade 10 months ago

Just try to trace the program for the given example. You will get to know that there can be a situation when there are multiple entries in the pri_Q having the same node(Because that node hasn't been explored yet, so the edges containing that node are been pushed to the Q). ?

So when finally one of those entries is popped and marked as explored there are still other entries in the Q having the same node. So while popping we again check whether that node is previously explored or not.

Just the trace the program for better understanding!

▲ 1 vote ● Reply ● Message ● Permalink



Vibhor Shukla 10 months ago

thanks...

▲ 0 votes ● Reply ● Message ● Permalink



Shiwang Gupta 5 months ago

Why is it giving wrong output while submitting?

```
#include <bits/stdc++.h>
#define PII pair<int,int>
#define PIP pair<int,pair<int,int>>
using namespace std;
struct disjointSet{
    struct Node{
        int data;
        Node* parent;
        int rank;
    };
    map<int,Node*> m;
    void makeSet(int data);
    void union2(int data1,int data2);
    int findSet(int data);
private:
    Node* findSet(Node* n);
};
void disjointSet::makeSet(int data){
    Node* n=new Node();
    n->data=data;
    n->parent=n;
    n->rank=0;
    m[data]=n;
}
void disjointSet::union2(int data1,int data2){
    Node* node1=m[data1];
    Node* node2=m[data2];
    Node* root1=findSet(node1);
    Node* root2=findSet(node2);
    if(root1->rank==root2->rank){
        root1->rank++;
        root2->parent=root1;
    }else if(root1->rank>root2->rank){
        root2->parent=root1;
    }else{
        root1->parent=root2;
    }
}
int disjointSet::findSet(int data){
    return findSet(m[data])->data;
}
disjointSet::Node* disjointSet::findSet(Node* n){
    while(n->parent!=n){
        n->parent=n->parent->parent;
        n=n->parent;
    }
    return n;
}
```

?

```

void addEdge(vector<PII> adj[],int a,int b,int w){
adj[a].push_back(make_pair(w,b));
//adj[b].push_back(make_pair(w,a));
}
struct compare{
bool operator()(PIP a,PIP b){
return (a.second).first>(b.second).first;
}
};
void kruskal(vector<PII> adj[],int n){
priority_queue<PIP,vector<PIP>,compare> pq;
for(int i=1;i<n+1;i++){
for(unsigned int j=0;j<adj[i].size();j++){
PII p=adj[i][j];
pq.push(make_pair(i,p));
}
}
disjointSet ds;
int min_sum=0;
int i=0;
while(i<n-1){
PIP p=pq.top();
pq.pop();
int x=p.first;
int w=(p.second).first;
int y=(p.second).second;
ds.makeSet(x);
ds.makeSet(y);
if(ds.findSet(x)==ds.findSet(y)) continue;
min_sum+=w;
ds.union2(x,y);
i++;
}
cout<<min_sum<<endl;
}
int main()
{
int t,n,m,a,b,w;
t=1;
while(t--){
cin>>n>>m;
vector<PII> adj[n+1];
while(m--){
cin>>a>>b>>w;
addEdge(adj,a,b,w);
}
kruskal(adj,n);
}
}

```

▲ 1 vote ● Reply ● Message ● Permalink



Pawan Kumar a month ago

All tutorials of Graphs are awesome. Use of STL in all the algorithms is quite impressive. Thanks!

▲ 1 vote ● Reply ● Message ● Permalink



Muskan Agarwal 2 years ago

memory limit exceeded??

▲ 0 votes ● Reply ● Message ● Permalink

?

**Junsang Dong** 2 years ago

Really helpful tutorials! Thank you hackerearth^^

▲ 0 votes ● Reply ● Message ● Permalink

**Shashank Shekhar** 2 years ago

Amazing tutorial. Thanks to the author :)

▲ 0 votes ● Reply ● Message ● Permalink

**Ajay Verma** a year ago

great tutorial....but..

i don't see the that when you are declaring pair globally, then why are you passing it into kruskal() function, there is no need of doing it...

▲ 0 votes ● Reply ● Message ● Permalink

**Mohammad Zaid** a year ago

yes u r right

▲ 0 votes ● Reply ● Message ● Permalink

**Kumar Saurabh** a year ago

What is wrong in it?

#include <iostream>

#include<algorithm>

#include<utility>

#include<vector>

using namespace std;

const int MAX = 1e4 + 5;

int nodes,edges,i;

pair<long long,pair<int,int>>p[MAX];

int id[MAX];

void initialize()

{

for(i=0;i<MAX;++i)

id[i]=i;

}

int root(int x)

{

while(id[x] != x)

{

id[x] = id[id[x]];

x = id[x];

}

return x;

}

void union1(int x, int y)

{

int p = root(x);

int q = root(y);

id[p] = id[q];

}

long long krushal(pair<long long,pair<int,int>>p[])

{

long long cost,mCost;

int x,y;

for(i=0;i<edges;i++)

{

x=p[i].second.first;

y=p[i].second.second;

?


```

cost=p[i].first;
if(root[x]!=root[y])
{
    mCost += cost;
    union1(x,y);
}
}
return mCost;
}

int main()
{
    long long minCost=0,wt=0;
    int x,y;
    cin>>nodes>>edges;
    initialize();
    for(i=0;i<edges;i++)
    {
        cin>>x>>y>>wt;
        p[i]= make_pair(wt,make_pair(x,y));
    }
    sort(p,p+edges);
    minCost=krushal(p);
    cout<<minCost;
    return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Sarthak Gupta 2 months ago

See the constraints!! M is 10^5 but you have taken $10^4 + 5$. That's the problem.

▲ 0 votes ● Reply ● Message ● Permalink



Anirudh Erabelly a year ago

Tried submitting but getting an error. I have put try catch block and got this as the output.

Unhandled Exception:

System.FormatException: Input string was not in a correct format.

at System.Number.StringToNumber (System.String str, NumberStyles options, System.NumberBuffer& number, System.Globalization.NumberFormatInfo info, Boolean parseDecimal) <0x7f5489fdebe0 + 0x000e0> in <filename unknown>:0

at System.Number.ParseInt32 (System.String s, NumberStyles style, System.Globalization.NumberFormatInfo info) <0x7f5489fdd2f0 + 0x00094> in <filename unknown>:0

at System.Int32.Parse (System.String s) <0x7f5489fbe670 + 0x0001d> in <filename unknown>:0

at System.Array.ConvertAll[TInput,TOutput] (System.TInput[] array, System.Converter`2 converter) <0x40e3be10 + 0x0006b> in <filename unknown>:0

at MyClass.Main (System.String[] args) <0x40e39d80 + 0x00100> in <filename unknown>:0

[ERROR] FATAL UNHANDLED EXCEPTION: System.FormatException: Input string was not in a correct format.

at System.Number.StringToNumber (System.String str, NumberStyles options, System.NumberBuffer& number, System.Globalization.NumberFormatInfo info, Boolean parseDecimal) <0x7f5489fdebe0 + 0x000e0> in <filename unknown>:0

at System.Number.ParseInt32 (System.String s, NumberStyles style, System.Globalization.NumberFormatInfo info) <0x7f5489fdd2f0 + 0x00094> in <filename unknown>:0

at System.Int32.Parse (System.String s) <0x7f5489fbe670 + 0x0001d> in <filename unknown>:0

at System.Array.ConvertAll[TInput,TOutput] (System.TInput[] array, System.Converter`2 converter) <0x40e3be10 + 0x0006b> in <filename unknown>:0

at MyClass.Main (System.String[] args) <0x40e39d80 + 0x00100> in <filename unknown>:0

▲ 0 votes ● Reply ● Message ● Permalink



Valentin Kostadinov a year ago

?

In the kruskal section, more specifically in the root function ,can't it be just if , instead of while. Nevertheless the cycle always makes only one iteration.

▲ 0 votes ● Reply ● Message ● Permalink



محمود عليو Edited a year ago

assume this id array:

element: 1,3,2,4,4,5

index:..... 0,1,2,3,4,5

and you call root(0), so you have to iterate two times to get the root of 0.

▲ 0 votes ● Reply ● Message ● Permalink



xarilaos Edited a year ago

3 steps:

copy code from prims algorithm, paste it in the editor window (be careful: language must be c++), submit.....

▲ 0 votes ● Reply ● Message ● Permalink



Sachin Kumar a year ago

why `id[x] = id[id[x]];`

???

▲ 0 votes ● Reply ● Message ● Permalink



abhi_sarkar96 a year ago

first head to Geeksforgeeks, understand union find operation, then it will be easy

▲ 0 votes ● Reply ● Message ● Permalink



Raghav Ravi Prakash a year ago

that's called path compression, makes the operation to find the root much faster

▲ 0 votes ● Reply ● Message ● Permalink



Rohit a year ago

Other than number of edges, is there any other hint i need to observe to use Prims over Kruskal? Please help

▲ 0 votes ● Reply ● Message ● Permalink



Leevansha Singla a year ago

Is there really need of any root function here ...??? I mean `id[x]` represents root here

▲ 0 votes ● Reply ● Message ● Permalink



SATYAJEET BEHERA a year ago

yep its necessary it makes sure that it doesn't form a cycle

▲ 0 votes ● Reply ● Message ● Permalink



Saurabh Sharma a year ago

following is an object oriented code in c++

```
#include <iostream>
```

```
#include <tuple>
```

```
#include <vector>
```

```
#include <numeric>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
typedef tuple<int, int, int> edge;
```

```
class union_find{
```

```
public:
```

```
int N;
```

```
vector<int> p, s;
```

```
union_find(int n){
```

?

```

p.resize(n);iota(p.begin(), p.end(), 0);
s.resize(n, 1);
N=n;
}
int find(int x);
void unite(int x, int y);
};
int union_find::find(int x){
return x==p[x]?x:find(p[x]);
}
void union_find::unite(int x, int y){
int a= find(x);
int b= find(y);
if(a<b)swap(a, b);
p[b] = a;s[a]+=s[b];N--;
}
int main()
{
int n, m;cin>>n>>m;
int ans=0;
vector<edge>edge_list(m);
while(m--){
int a, b, c;cin>>a>>b>>c;
edge_list.push_back(make_tuple(a, b, c));
}
sort(edge_list.begin(), edge_list.end(),[](const edge &a, const edge &b){
return get<2>(a) < get<2>(b);
});
union_find set(n); int i=0;
while(set.N>1){
int q = get<0>(edge_list[i]);
int r = get<1>(edge_list[i]);
if(set.find(q)!=set.find(r)){
set.unite(q, r);
ans+=get<2>(edge_list[i]);
}
i++;
}
cout<<ans;
return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Pallav Keshari a year ago

the two lines .

y=adj[i][j] & if (vis[j]==0) are not required at all .

▲ 0 votes ● Reply ● Message ● Permalink



Kunal Gupta a year ago

Can anyone explain the root function??

Else I've understood

▲ 0 votes ● Reply ● Message ● Permalink



SATYAJEET BEHERA a year ago

root function finds the root node of the tree;

▲ 0 votes ● Reply ● Message ● Permalink



Nitin Rathodiya a year ago

?

well it is actually for finding a representative of a set lets suppose i have a set with index 1, 2, 3 and their respective ids are 2 2 4 it means in a set only one representative available where value of $x=id[x]$. in this case $2=id[2]$.so Representative is 2. Root function is intended for this task.

▲ 0 votes ● Reply ● Message ● Permalink



Neha Poonia a year ago

a very good explanation of disjoint set is given

<https://www.hackerearth.com/practice/notes/disjoint-set-union-union-find/> here.

▲ 0 votes ● Reply ● Message ● Permalink



wyrllvillazorda a year ago

give me another example of prim's algorithm using Java.. Is it like Breadth First Search?

▲ 0 votes ● Reply ● Message ● Permalink



Raghav Ravi Prakash a year ago

it is except instead of using a queue, we use priority queue giving the lowest weights higher priority. So rather than just dequeuing the node that comes first, we dequeue the node having the lowest weight.

▲ 0 votes ● Reply ● Message ● Permalink



Raghav Ravi Prakash a year ago

Here's prim's algo in java

```
import java.util.*;

class Pair
{
    public int node;
    public int weight;
    public Pair(int node, int weight)
    {
        this.node = node;
        this.weight = weight;
    }
}

class PriorityQueueComparator implements Comparator<Pair>
{
    @Override
    public int compare(Pair pair1, Pair pair2)
    {
        int weight1 = pair1.weight;
        int weight2 = pair2.weight;
        if(weight1 < weight2)
            return -1;
        if(weight1 > weight2)
            return 1;

        return 0;
    }
}

class PrimAlgorithm
{
    private ArrayList<Pair>[] adjacencyList;
    private int numNodes;
    public PrimAlgorithm(int N)
    {
        numNodes = N;
        adjacencyList = new ArrayList[N+1];
    }
}
```

?

```
public int computeMinWeight(Scanner in, int M)
{
    initializeList();
    while(M > 0)
    {
        int node1 = in.nextInt();
        int node2 = in.nextInt();
        int weight = in.nextInt();
        updateList(node1,node2,weight);
        M--;
    }
    in.close();
    return getWeightOfMST();
}

private void initializeList()
{
    for(int i = 1; i <= numNodes; i++)
        adjacencyList[i] = new ArrayList<Pair>();
}

private void updateList(int node1, int node2, int weight)
{
    Pair pairNode1 = new Pair(node2,weight);
    Pair pairNode2 = new Pair(node1,weight);
    adjacencyList[node1].add(pairNode1);
    adjacencyList[node2].add(pairNode2);
}

private int getWeightOfMST()
{
    Comparator<Pair> pqCompare = new PriorityQueueComparator();
    PriorityQueue<Pair> pq = new PriorityQueue<Pair>(100000,pqCompare);
    boolean[] visited = new boolean[numNodes+1];
    Arrays.fill(visited,1,visited.length,false);
    Pair startPair = new Pair(1,0);
    pq.add(startPair);
    int minWeight = 0;
    while(pq.size() > 0)
    {
        Pair pqPair = (Pair) pq.poll();
        int sourceNode = pqPair.node;
        int weight = pqPair.weight;

        if(visited[sourceNode] == true)
            continue;
        minWeight += weight;
        visited[sourceNode] = true;
        for(int i = 0; i < adjacencyList[sourceNode].size(); i++)
        {
            Pair destPair = (Pair) adjacencyList[sourceNode].get(i);
            int destNode = destPair.node;
            if(visited[destNode] == false)
                pq.add(destPair);
        }
    }
    return minWeight;
}
}
```

?

```

public class MSTPrim
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int N = in.nextInt();
        int M = in.nextInt();
        PrimAlgorithm pa = new PrimAlgorithm(N);
        int minWeight = pa.computeMinWeight(in,M);
        System.out.println(minWeight);
    }
}

```

▲ 0 votes ● Reply ● Message ● Permalink



wyrllvillazorda a year ago

thnk you..

▲ 0 votes ● Reply ● Message ● Permalink



Raghav Ravi Prakash a year ago

Runtime error while submitting. Please help.

```

import java.util.*;
class Pair
{
    public int sourceNode;
    public int destNode;
    public Pair(int source, int dest)
    {
        sourceNode = source;
        destNode = dest;
    }
}
class ArrayListObject
{
    private int weight;
    private Pair pair;
    public ArrayListObject(int weight, Pair pair)
    {
        this.weight = weight;
        //this.pair = pair;
    }
    int sourceNode = pair.sourceNode;
    int destNode = pair.destNode;
    this.pair = new Pair(sourceNode,destNode);
}
public int getWeight()
{
    return weight;
}
public Pair getPair()
{
    return pair;
}
}
class SortList implements Comparator<ArrayListObject>
{
    @Override
    public int compare(ArrayListObject obj1, ArrayListObject obj2)
    {

```

?

```

int weight1 = obj1.getWeight();
int weight2 = obj2.getWeight();
if(weight1 <= weight2)
return -1;
return 1;
}
}

public class MSTKruskal
{
public static ArrayList<ArrayListObject> list = new ArrayList<ArrayListObject>();
public static void main(String[] args)
{
Scanner in = new Scanner(System.in);
int N = in.nextInt();
int M = in.nextInt();
int[] rootArray = new int[N+1];
int[] size = new int[N+1];
initializeRoots(rootArray,N);
Arrays.fill(size,1,size.length,1);
while(M > 0)
{
int node1 = in.nextInt();
int node2 = in.nextInt();
int weight = in.nextInt();
initializeList(node1,node2,weight);
M--;
}
SortList obj = new SortList();
Collections.sort(list,obj);
int sumWeightsMST = computeSum(rootArray,size);
System.out.println(sumWeightsMST);
}

private static void initializeRoots(int[] rootArray, int N)
{
for(int i = 1; i <= N; i++)
rootArray[i] = i;
}

private static void initializeList(int node1, int node2, int weight)
{
Pair pair = new Pair(node1,node2);
ArrayListObject obj = new ArrayListObject(weight,pair);
list.add(obj);
}

private static int computeSum(int[] rootArray, int[] size)
{
int minimumWeight = 0;
for(int i = 0; i < list.size(); i++)
{
ArrayListObject obj = (ArrayListObject) list.get(i);
int weight = obj.getWeight();
Pair pair = obj.getPair();
int node1 = pair.sourceNode;
int node2 = pair.destNode;
int root1 = getRoot(rootArray,node1);
int root2 = getRoot(rootArray,node2);
if(root1 != root2) // doesn't form a cycle
{

```

?

```

minimumWeight += weight;
union(root1,root2,rootArray,size);
}
}
return minimumWeight;
}
private static int getRoot(int[] rootArray, int i)
{
while(i != rootArray[i])
i = rootArray[rootArray[i]];
return i;
}
private static void union(int root1, int root2, int[] rootArray, int[] size)
{
if(size[root1] < size[root2])
{
rootArray[root1] = rootArray[root2];
size[root2] += size[root1];
}
else
{
rootArray[root2] = rootArray[root1];
size[root1] += size[root2];
}
}
}
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Hemant Mangwani a year ago

/*

Written by:- Hemant Kumar Mangwani on 03/08/17

references :-

<https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/tutorial/>

<https://www.youtube.com/watch?v=fAuF0EuZVck&t=423s>

<https://www.youtube.com/watch?v=tKwnms5iRBU&t=2054s>

<https://www.youtube.com/watch?v=71UQH7Pr9kU>

<http://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/>

*/

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
ll a[100000001];
vector<pair<ll,pair<ll,ll>>> v;
ll root(ll i)
{
while(a[i]!=i)
{
a[i]=a[a[i]];
i=a[i];
}
return i;
}
void unions(ll x,ll y)
{
a[root(x)]=a[root(y)];
}
ll kruskals(ll e)
{

```

?


```

ll x,y,w,mc=0,i;
for(i=0;i<e;i++)
{
x=v[i].second.first;
y=v[i].second.second;
w=v[i].first;
if(root(x)!=root(y))
{
unions(x,y);
mc+=w;
}
}
return mc;
}

//driver
int main()
{
ll n,e,i;
cout<<"Enter no of nodes and edges\n";
cin>>n>>e;
//Initilaization
for(i=0;i<n;i++) a[i]=i;
for(i=0;i<e;i++)
{
ll x,y,w;
cin>>x>>y>>w;
v.push_back(make_pair(w,make_pair(x,y)));
}
sort(v.begin(),v.end());
cout<<kruskals(e)<<"\n";
return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Vikas Yadav Edited a year ago

why it is written as long long only not any int...or long long int since long long int is a data type ..not long long

▲ 0 votes ● Reply ● Message ● Permalink



rahul patel a year ago

same here..Why??

▲ 0 votes ● Reply ● Message ● Permalink



Sushil Temp a year ago

Help please explain why my code is giving wrong answer

```

import java.util.*;
class TestClass {
public static Comparator<Node> com = new Comparator<Node>(){

@Override
public int compare(Node c1, Node c2) {
return (int) (c1.w - c2.w);
}
};
public static void main(String args[] ) throws Exception {
Scanner s = new Scanner(System.in);
int n = s.nextInt();
int m= s.nextInt();
Node [] nd=new Node[n];

```

?

```

for (int i = 0; i < n; i++) {
    nd[i]=new Node();
}
for (int i = 0; i < m; i++) {
    int y=s.nextInt();
    int u= s.nextInt();
    int h=s.nextInt();
    nd[y-1].ar.add(new Pair(u-1,h));
    nd[u-1].ar.add(new Pair(y-1,h));
}

Queue<Node> q = new PriorityQueue<>(com);
nd[0].w=0;
q.add(nd[0]);
long cost=0;
while(!q.isEmpty()){
    Node k=q.poll();
    if(k.v){
        continue;
    }

    // System.out.println("adding cost = "+k.w);
    cost=cost+k.w;
    k.v=true;
    // System.out.println("k.ar.size() "+k.ar.size());
    for(int i=0;i<k.ar.size();i++){
        Pair p=k.ar.get(i);
        //System.out.println("p.n = "+p.n+" nd[p.n].w "+nd[p.n].w+" p.w = "+p.w);
        if(nd[p.n].v)
            continue;
        if(nd[p.n].w>p.w)
            nd[p.n].w=p.w;

        q.add(nd[p.n]);
    }
}
System.out.println(cost);

}

}
class Node{
    // int w;
    int w=Integer.MAX_VALUE;
    boolean v;
    ArrayList<Pair> ar=new ArrayList<Pair>();
}
class Pair{
    int w ;
    int n;
    Pair(int r,int q)
    {
        w=q;
        n=r;
    }
}

```

▲ 0 votes ● Reply ● Message ● Permalink



PAVAN KUMAR YADAV a year ago

?

CAN ANYONE HELP TELLING ME WHATS WRONG IN THIS CODE. IT IS GIVING WRONG SOLUTION

```
//PRIM's ALGORITHM
#include<bits/stdc++.h>
using namespace std;
const int MAX=1e4+5;
typedef pair<long long,long long> p;
bool mark[MAX];
vector<p> adj[MAX];
long long minm(long long x)
{
    priority_queue<p,vector<p>,greater<p> > Q;
    p p1;
    long long mcost=0;
    Q.push(make_pair(x,0));
    while(!Q.empty())
    {
        p1=Q.top();
        Q.pop();
        x=p1.first;
        if(mark[x]==true)
            continue;
        mcost+=p1.second;
        mark[x]=true;
        for(long long i=0;i<adj[x].size();i++)
        {
            if(mark[adj[x][i].first]==false)
                Q.push(adj[x][i]);
        }
    }
    return mcost;
}
int main()
{
    long long nodes,edges;
    long long x,y,weight;
    cin>>nodes>>edges;
    for(long long i=0;i<edges;i++)
    {
        cin>>x>>y>>weight;
        adj[x].push_back(make_pair(y,weight));
        adj[y].push_back(make_pair(x,weight));
    }
    cout<<minm(1);
    return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink



Harsh Goyal 10 months ago

```
if(mark[adj[x][i].first]==false)
    Q.push(adj[x][i]);
```

Q is a queue of pair<int,int> bt we are trying to push one argument..Thats why getting aan error.
:)

▲ 0 votes ● Reply ● Message ● Permalink



md Sazid Khan a year ago

time complexity for Kruskal's Algorithm
will be $O(E \log E + E)$ or $O(E \log E)$, correct me if i'm wrong

▲ 0 votes ● Reply ● Message ● Permalink

Akash Garg a year ago



can anyone explain root in kruskal's algo?

▲ 0 votes ● Reply ● Message ● Permalink



Amar Anand 10 months ago

can the edges in a MST be disconnected?

▲ 0 votes ● Reply ● Message ● Permalink



Vedant Gawade Edited 10 months ago

here we are computing the MST on a single connected graph say G, therefore the MST computed will include all the nodes(not edges) of graph G. Hence the selected edges will always be connected to form a tree.

Another case: If you are computing MST on a number of disconnected components of a graph, then obviously the MST's generated will also be disconnected.

And one more thing, even if the edges in a MST formed from a single component are disconnected, then it violates the basic property of spanning trees(ie: In a spanning tree you must be able to reach any node from a given point)

▲ 0 votes ● Reply ● Message ● Permalink



Vedant Gawade 10 months ago

Great tutorial! Thanks a lot.

▲ 0 votes ● Reply ● Message ● Permalink



Nashrah Maeraj 10 months ago

how to overcome run time error?

▲ 0 votes ● Reply ● Message ● Permalink



rushikesh fanse 10 months ago

kruskal is smooth :)

▲ 0 votes ● Reply ● Message ● Permalink



Amit Kumar 8 months ago

kruskals algo is giving wrong ans when you just copy it and paste in above question , so there is some error , please rectify it

▲ 0 votes ● Reply ● Message ● Permalink



Vishal Yadav 8 months ago

how to find second minimum spanning tree??

▲ 0 votes ● Reply ● Message ● Permalink



Sagar Patil 7 months ago

nice artical

▲ 0 votes ● Reply ● Message ● Permalink



Manveer Singh 7 months ago

But your spellings aren't nice!

▲ 0 votes ● Reply ● Message ● Permalink



Ravi Kumar Tahlani 7 months ago

C++ working program

```
=====
#include<bits/stdc++.h>
#define ll long long int
using namespace std;
void initialize(int id[],int n){
for(int i = 1;i <= n; i++){
id[i] = i;
}
}
```

?

```

return;
}
int root(int a,int id[]){
while(id[a] != a){
a = id[ id[a] ];
}
return a;
}
void join(int a, int b, int id[]){
int root_a = root(a,id);
int root_b = root(b,id);

if(root_a != root_b){
id[ root_a ] = id[ root_b ];
}

return;
}
ll kruskal(pair< ll , pair< int , int > > p[], int id[], int n){

ll min_weight = 0;
ll a,b,w,root_a,root_b;
for(int i = 1; i <= n ; i++){

a = p[i].second.first;
b = p[i].second.second;
w = p[i].first;
root_a = root(a,id);
root_b = root(b,id);

if( root_a != root_b ){
min_weight += w;
join(a,b,id);

}
}

return min_weight;
}
int main(){

int n,m,a,b,w;
cin>>n>>m;

int *id = new int[n+1];
initialize(id,n);
pair< ll, pair< int, int > > p[m+1];
for(int i = 1;i <= m; i++){
cin>>a>>b>>w;
p[i].first = w;
p[i].second.first = a;
p[i].second.second = b;
}

sort( p , p+(m+1) );

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

```

?

```
cout<<"( "<<p[i].first <<" , "<<"<"<<p[i].second.first<<" , "<<p[i].second.second<<"> "<<" ) ";
}*/
ll mini = kruskal( p , id , m);
cout<<mini;
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink



ky lee 6 months ago

Can someone check my Python code --> It give me wrong answer, but right in simple test

```
import Queue
def prim(edge_dict, n):
    can_edge = Queue.PriorityQueue()
    tree_ver = set()
    tree_ver.add(1)
    for v in edge_dict[1]:
        if v not in tree_ver:
            can_edge.put((edge_dict[1][v], 1, v))

    value = 0
    while can_edge.qsize() > 0:
        (w,a,b) = can_edge.get()
        if b not in tree_ver: # and a in tree_ver:
            tree_ver.add(b)
            value += w
            for x in edge_dict[b]:
                if x not in tree_ver:
                    can_edge.put((edge_dict[b][x],b,x))

    print value
    edge_dict = {}
    [n,m] = [long(x) for x in raw_input().split()]
    for i in range(0, m):
        [a,b,w] = [long(x) for x in raw_input().split()]
        if a in edge_dict:
            edge_dict[a][b] = w
        else:
            edge_dict[a] = {b:w}

        if b in edge_dict:
            edge_dict[b][a] = w
        else:
            edge_dict[b] = {a:w}

    #print edge_dict
    prim(edge_dict, n)
```

▲ 0 votes ● Reply ● Message ● Permalink



nimesh jain 5 months ago

```
#include<bits/stdc++.h>
#define mp make_pair
#define pb push_back
#define fi first
#define se second
#define ll long long
using namespace std;
ll arr[100005];
ll size[100005];
```

?

```

ll n,m;
pair<ll,pair<ll,ll> >p[100005];
ll root(ll w){
while(arr[w]!=w)
w=arr[w];
return w;
}
void union1(ll x,ll y){
ll p=root(x);
ll q=root(y);
if(size[p]<size[q]){
arr[p]=arr[q];
size[q]+=size[p];
}
else{
arr[q]=arr[p];
size[p]+=size[q];
}
}
ll kruskl(pair<ll,pair<ll,ll> >p[]){
ll mco=0;
ll i;
for(i=0;i<m;i++){
ll x=p[i].se.fi;
ll y=p[i].se.se;
if(root(x)!=root(y)){
mco+=p[i].fi;
union1(x,y);
}
}
return mco;
}
int main(){

ll i,a,b,c;
for(i=0;i<100005;i++){
arr[i]=i;
size[i]=1;
}
cin>>n>>m;
for(i=0;i<m;i++){
cin>>a>>b>>c;
p[i]=mp(c,mp(a,b));
}
sort(p,p+m);
ll min=kruskl(p);
cout<<min<<endl;
return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Shubham Jawandhiya 3 months ago

What is the use of line `id[x]=id[id[x]]` necessary? Because the code is accepted even without it.

▲ 0 votes ● Reply ● Message ● Permalink



Adit Chanchal Edited 3 months ago

Yes, the code will even run without it.

The above line is used only for path compression which makes the code run faster.

▲ 0 votes ● Reply ● Message ● Permalink

?



Rishabh Agrawal 3 months ago

Why my code is giving wrong answer can anyone explain me?

```
#include <bits/stdc++.h>
using namespace std;
#define f(i,a,b) for(long long i=a;i<b;i++)
#define s(c) scanf("%lld",&c)
#define p(m) printf("%lld\n",m)
#define pb push_back
#define MOD 1000000007
#define boost ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL)
typedef long l;
typedef long long ll;
typedef pair<int,int> Pl;
pair<int,Pl> p[100000];
int parent[10001];
int n,m;
int find(int x)
{
    if(parent[x]==x)
        return x;
    else
        return parent[x]=find(parent[parent[x]]);
}
void union1(int x,int y)
{
    x=find(x);
    y=find(y);
    parent[x]=y;
}
int kruskal(pair<int,Pl> p[100000])
{
    int minWt=0;

    for(int i=0;i<n;i++)
    {
        int x=p[i].second.first;
        int y=p[i].second.second;
        int wt=p[i].first;
        if(find(x)!=find(y))
        {
            //cout<<"weight="<<wt<<endl;
            minWt+=wt;
            union1(x,y);
        }
    }
    return minWt;
}
int main()
{
    boost;

    int minWt;
    cin>>n>>m;

    for(int i=0;i<m;i++)
    {
```

?


```

int u,v,wt;
cin>>u>>v>>wt;
p[i]=make_pair(wt,make_pair(u,v));

}
for(int i=1;i<=n;i++)
parent[i]=i;
sort(p,p+m);

minWt=kruskal(p);

cout<<minWt<<endl;

return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Adit Chanchal 3 months ago

simple explanation and easy to understand.
loved it <3

▲ 0 votes ● Reply ● Message ● Permalink



ak389 3 months ago

Q.push(make_pair(0, x));
in prim code why we are taking the weight as 0 for first input ???

▲ 0 votes ● Reply ● Message ● Permalink



mohammed_almasry 3 months ago

```

<<<<<<<Accepted>>>>>>>
///In the name of Allah
///BY_Mohammed_ALmasry
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
///double rad= ( ( Angle_in_degree *pi ) * 1.0 ) / 180;
///tan(rad):only in radians
#define Fast ios_base::sync_with_stdio(false)
#define pi 3.14159265358979323846//pi=acos(-1)
#define read freopen("input.txt","r",stdin)
#define write freopen("output.txt","w",stdout)
#define fi first
#define se second
#define pb push_back
#define f0(i,n) for(int i=0;i<n;i++)
#define f1(i,n) for(int i=1;i<=n;i++)
#define ff(i,s,e) for(int i=s;i<=e;i++)
#define all(v) v.begin(),v.end()
#define mod 1000000007
#define sz size
#define pii pair<int,int>
#define pll pair<ll,ll>
//cout<<fixed<<setprecision(0);
int xx[9]={000,-1,1,0,0,-1,-1,1,1};
int yy[9]={000,0,0,1,-1,1,-1,1,-1};
int hx[8]={ 1, -1, 2, -2, 2, -2, 1, -1 };
int hy[8]={ 2, 2, 1, 1, -1, -1, -2, -2 };
int kx[8]={1,1,0,-1,1};
int ky[8]={1,0,1,0,0};

```

?

```

bool sieve[1000000];
vector< pii >v[100100];
bool vi[100100];
ll mst(int s)
{
    ll mst_sum=0;
    priority_queue<pii,vector<pii>,greater<pii>>pq;
    pq.push({0,s});
    while(!pq.empty())
    {
        pii topp=pq.top();
        pq.pop();
        if(vi[topp.se])continue;
        vi[topp.se]=1;
        mst_sum+=topp.fi;
        for(int i=0;i<v[topp.se].size();i++)
        {
            pii son=v[topp.se][i];
            if(vi[son.se]==0)
                pq.push(son);
        }
    }
    return mst_sum;
}
int main()
{
    Fast;
    int n,m;
    cin>>n>>m;
    while(m--)
    {
        int x,y,z;
        cin>>x>>y>>z;
        v[x].pb({z,y});
        v[y].pb({z,x});
    }
    cout<<mst(1);
}

```

▲ 0 votes ● Reply ● Message ● Permalink



aditya raj 3 months ago

simplest solution, just coded the pseudo code:

```

#include <bits/stdc++.h>
using namespace std;
#define mod 1000000007
#define boost ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
#define size 10004
int p[10005];
int rk[10005];
struct EDGE{
    int s,e,w;
};
bool cmp(EDGE a,EDGE b)
{
    return a.w<b.w;
}
void make_set(int n)
{
    for(int i=1;i<=n;i++)
    {

```

?

```

p[i]=i;
rk[i]=0;
}
}
int find_set(int x)
{
if(p[x]!=x)
p[x]=find_set(p[x]);
return p[x];
}
void union_set(int a,int b)
{
int x,y;
x=find_set(a);
y=find_set(b);
if(rk[x]<rk[y])
{ p[x]=y;}
else{
p[y]=x;
if(rk[x]==rk[y])
rk[x]++;
}
}
int main()
{

boost
#ifndef ONLINE_JUDGE
freopen("input.txt","r",stdin);
freopen("output.txt","w",stdout);
#endif
int n,m;
cin>>n>>m;
EDGE edge[m];
for(int i=0;i<m;i++)
{
cin>>edge[i].s>>edge[i].e>>edge[i].w;
}
sort(edge,edge+m,cmp);
int c=0,sum=0;
int i=0;
make_set(n);
while(c<=n-1 && i<m)
{
int a=find_set(edge[i].s);
int b=find_set(edge[i].e);
if(a!=b){
union_set(a,b);
c++;
sum+=edge[i].w;
}
i++;
}
cout<<sum<<"\n";
return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Atul Verma a month ago

one of the best tutorials !!

?

▲ 0 votes ● Reply ● Message ● Permalink



Amit Ranjan a month ago

how can i print the path which is used in making mst?

▲ 0 votes ● Reply ● Message ● Permalink



Mudassar Raza 5 days ago

I think in the function: `kruskal ()` writtten above, there is no need to pass any parameter as `p[MAX]` is a global variable.

correct me if I missed something

▲ 0 votes ● Reply ● Message ● Permalink

[About Us](#)

[Innovation Management](#)

[Technical Recruitment](#)

[University Program](#)

[Developers Wiki](#)

[Blog](#)

[Press](#)

[Careers](#)

[Reach Us](#)



Site Language: [English](#) ▼ | [Terms and Conditions](#) | [Privacy](#) | © 2018 HackerEarth

