

# Dense Graph:

## Kruskal's Algorithm

```
test_kruskal.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 struct Edge
6 { int u, v, weight; };
7
8 void makeSet(int V, vector<int>& parent, vector<int>& rank)
9 {
10     for(int i=0; i<V; i++)
11     {
12         parent[i] = i;
13         rank[i] = 0;
14     }
15 }
16 int findSet(int node, vector<int>& parent)
17 {
18     if(parent[node] != node)
19         parent[node] = findSet(parent[node], parent);
20     return parent[node];
21 }
22 void UnionSet(int u, int v, vector<int>& parent, vector<int>& rank)
23 {
24     int rootU = findSet(u, parent);
25     int rootV = findSet(v, parent);
26
27     if(rootU != rootV)
28     {
29         if(rank[rootU] >= rank[rootV])
30             parent[rootV] = rootU;
31         else if(rank[rootU] < rank[rootV])
32             parent[rootU] = rootV;
33         else
34             rank[rootU]++;
35     }
36 }
37
38 void kruskal(int V, vector<vector<int>>& adjMatrix)
39 {
40     vector<int> parent(V);
41     vector<int> rank(V);
42     vector<Edge> edges;
43     vector<Edge> mstEdges;
44     int totalWeight = 0;
45
46     for(int i=0; i<V; i++)
47     {
48         for(int j=i+1; j<V; j++)
49         {
50             if(adjMatrix[i][j] != 0)
51                 edges.push_back({i, j, adjMatrix[i][j]});
52         }
53     }
54     sort(edges.begin(), edges.end(), [](const Edge& a, const Edge& b)
55         {return a.weight < b.weight;});
56     makeSet(V, parent, rank);
57
58     Edge Weights
59     2 - 6      1
60     4 - 8      1
61     1 - 4      2
62     5 - 6      2
63     4 - 7      2
64     1 - 5      3
65     0 - 1      3
66     0 - 3      4
67     8 - 9      7
68     25
69     7.083 ms
70
71     Process returned 0 (0x0)   execution time : 0.138 s
72     Press any key to continue.
```

```
test_kruskal.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X
34     rank[rootU]++;
35 }
36 }
37 void kruskal(int V, vector<vector<int>>& adjMatrix)
38 {
39     vector<int> parent(V);
40     vector<int> rank(V);
41     vector<Edge> edges;
42     vector<Edge> mstEdges;
43     int totalWeight = 0;
44
45     for(int i=0; i<V; i++)
46     {
47         for(int j=i+1; j<V; j++)
48         {
49             if(adjMatrix[i][j] != 0)
50                 edges.push_back({i, j, adjMatrix[i][j]});
51         }
52     }
53     sort(edges.begin(), edges.end(), [](const Edge& a, const Edge& b)
54         {return a.weight < b.weight;});
55     makeSet(V, parent, rank);
56
57     for(Edge edge: edges)
58     {
59         int u = edge.u;
60         int v = edge.v;
61         if(findSet(u, parent) != findSet(v, parent))
62         {
63             UnionSet(u, v, parent, rank);
64             mstEdges.push_back(edge);
65             totalWeight += edge.weight;
66         }
67     }
68
69     cout<<"Edge      Weight<<endl;
70
71     for(Edge edge: mstEdges)
72     {
73         cout<<edge.u<<" - "<<edge.v<<"      "<<edge.weight<<endl;
74     }
75     cout<<totalWeight<<endl;
76 }
77
78 int main()
79 {
80     vector<vector<int>> graph = {
81         // provided dense graph
82         {0,3,0, 4,4,0,0, 0,0,0},
83         {3,0,10, 0,2,3,0, 0,0,0},
84         {0,10,0, 0,0,6,1, 0,0,0},
85
86         {4,0,0, 0,5,0,0, 6,0,0},
87         {4,2,0, 5,0,11,0, 2,1,0},
88         {0,3,6, 0,11,0,2, 0,3,11},
89         {0,0,1, 0,0,2,0, 0,0,8},
90
91         {0,0,0, 6,2,0,0, 0,4,0},
92         {0,0,0, 0,1,3,0, 4,0,7},
93         {0,0,0, 0,0,11,8, 0,7,0},
94     };
95
96     auto Start = chrono::high_resolution_clock::now();
97     kruskal(graph.size(), graph);
98     auto End = chrono::high_resolution_clock::now();
99     chrono::duration<double, milli> duration = End - Start;
100
101     cout<<duration.count()<<" ms"<<endl;
102
103     return 0;
104 }
```

# Prim's Algorithm

```
test_primes.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 void prim(vector<vector<int>>& graph, int start=0)
6 {
7     int V = graph.size();
8     vector<int> key(V, INT_MAX);
9     vector<int> parent(V, -1);
10    vector<bool> inMst(V, false);
11    vector<pair<int,int>> mstEdges;
12    priority_queue<pair<int,int>, vector<pair<int,int>>, greater<pair<int,int>>> minHeap;
13    int totalWeight = 0;
14
15    key[start] = 0;
16    minHeap.push({0, start});
17
18    while(!minHeap.empty())
19    {
20        int u = minHeap.top().second;
21        int weight = minHeap.top().first;
22        minHeap.pop();
23
24        if(inMst[u]) continue;
25
26        inMst[u] = true;
27        totalWeight += weight;
28
29        if(parent[u] != -1)
30            mstEdges.push_back({parent[u], u});
31
32        for(int v=0; v<V; v++)
33        {
34            if(!inMst[v] && graph[u][v]!=0&&graph[u][v]<key[v])
35            {
36                key[v] = graph[u][v];
37            }
38        }
39    }
40
41    cout<<"Edges    Weight"<<endl;
42    for(auto edge: mstEdges)
43    {
44        cout<<edge.first<<" - "<<edge.second<<"    "<<graph[edge.first][edge.second]<<endl;
45    }
46    cout<<"Total weight:"<<totalWeight<<endl;
47
48    int main()
49    {
50        vector<vector<int>> graph = {
51            // provided dense graph
52            {0,3,0, 4,4,0,0, 0,0,0},
53            {3,0,10, 0,2,3,0, 0,0,0},
54            {0,10,0, 0,0,6,1, 0,0,0},
55
56            {4,0,0, 0,5,0,0, 6,0,0},
57            {4,2,0, 5,0,11,0, 2,1,0},
58            {0,3,6, 0,11,0,2, 0,3,11},
59            {0,0,1, 0,0,2,0, 0,0,8},
60
61            {0,0,0, 6,2,0,0, 0,4,0},
62            {0,0,0, 0,1,3,0, 4,0,7},
63            {0,0,0, 0,0,11,8, 0,7,0},
64        };
65
66        auto Start = chrono::high_resolution_clock::now();
67        prim(graph);
68        auto End = chrono::high_resolution_clock::now();
69        chrono::duration<double, milli> duration = End - Start;
70        cout<<duration.count()<<" mS"<<endl;
71        return 0;
72    }
73 }
```

```
31
32 for(int v=0; v<V; v++)
33 {
34     if(!inMst[v] && graph[u][v]!=0&&graph[u][v]<key[v])
35     {
36         key[v] = graph[u][v];
37         parent[v] = u;
38         minHeap.push({graph[u][v], v});
39     }
40 }
41
42 cout<<"Edges    Weight"<<endl;
43 for(auto edge: mstEdges)
44 {
45     cout<<edge.first<<" - "<<edge.second<<"    "<<graph[edge.first][edge.second]<<endl;
46 }
47 cout<<"Total weight:"<<totalWeight<<endl;
48
49 Process returned 0 (0x0)   execution time : 0.139 s
50 Press any key to continue.
```

C:\Users\Golam Mostafa Rabby\OneDrive\B.Sc in Duet\3rd year 1st semester\Sessiona... C/C++ Windows (CR+LF) WINDOWS-1252 Line 30, Col 33, 9:44 PM 12/29/2024

```
test_primes.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X
42 cout<<"Edges    Weight"<<endl;
43 for(auto edge: mstEdges)
44 {
45     cout<<edge.first<<" - "<<edge.second<<"    "<<graph[edge.first][edge.second]<<endl;
46 }
47 cout<<"Total weight:"<<totalWeight<<endl;
48
49 int main()
50 {
51     vector<vector<int>> graph = {
52         // provided dense graph
53         {0,3,0, 4,4,0,0, 0,0,0},
54         {3,0,10, 0,2,3,0, 0,0,0},
55         {0,10,0, 0,0,6,1, 0,0,0},
56
57         {4,0,0, 0,5,0,0, 6,0,0},
58         {4,2,0, 5,0,11,0, 2,1,0},
59         {0,3,6, 0,11,0,2, 0,3,11},
60         {0,0,1, 0,0,2,0, 0,0,8},
61
62         {0,0,0, 6,2,0,0, 0,4,0},
63         {0,0,0, 0,1,3,0, 4,0,7},
64         {0,0,0, 0,0,11,8, 0,7,0},
65     };
66
67     auto Start = chrono::high_resolution_clock::now();
68     prim(graph);
69     auto End = chrono::high_resolution_clock::now();
70     chrono::duration<double, milli> duration = End - Start;
71     cout<<duration.count()<<" mS"<<endl;
72     return 0;
73 }
```

```
31
32 for(int v=0; v<V; v++)
33 {
34     if(!inMst[v] && graph[u][v]!=0&&graph[u][v]<key[v])
35     {
36         key[v] = graph[u][v];
37         parent[v] = u;
38         minHeap.push({graph[u][v], v});
39     }
40 }
41
42 cout<<"Edges    Weight"<<endl;
43 for(auto edge: mstEdges)
44 {
45     cout<<edge.first<<" - "<<edge.second<<"    "<<graph[edge.first][edge.second]<<endl;
46 }
47 cout<<"Total weight:"<<totalWeight<<endl;
48
49 Process returned 0 (0x0)   execution time : 0.139 s
50 Press any key to continue.
```

C:\Users\Golam Mostafa Rabby\OneDrive\B.Sc in Duet\3rd year 1st semester\Sessiona... C/C++ Windows (CR+LF) WINDOWS-1252 Line 75, Col 41, 9:45 PM 12/29/2024

# Sparse Graph:

## Kruskal's Algorithm

```
test_kruskal.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 struct Edge
6 { int u, v, weight; };
7
8 void makeSet(int V, vector<int>& parent, vector<int>& rank)
9 {
10     for(int i=0; i<V; i++)
11     {
12         parent[i] = i;
13         rank[i] = 0;
14     }
15 }
16
17 int findSet(int node, vector<int>& parent)
18 {
19     if(parent[node] != node)
20         parent[node] = findSet(parent[node], parent);
21     return parent[node];
22 }
23
24 void UnionSet(int u, int v, vector<int>& parent, vector<int>& rank)
25 {
26     int rootU = findSet(u, parent);
27     int rootV = findSet(v, parent);
28
29     if(rank[rootU] > rank[rootV])
30         parent[rootV] = rootU;
31     else if(rank[rootU] < rank[rootV])
32         parent[rootU] = rootV;
33     else
34         rank[rootU]++;
35 }
36
37 void kruskal(int V, vector<vector<int>>& adjMatrix)
38 {
39     vector<int> parent(V);
40     vector<int> rank(V);
41     vector<Edge> edges;
42     vector<Edge> mstEdges;
43     int totalWeight = 0;
44
45     for(int i=0; i<V; i++)
46     {
47         for(int j=i+1; j<V; j++)
48         {
49             if(adjMatrix[i][j] != 0)
50                 edges.push_back({i, j, adjMatrix[i][j]});
51         }
52     }
53     sort(edges.begin(), edges.end(), [](const Edge& a, const Edge& b)
54         {return a.weight < b.weight;});
55     makeSet(V, parent, rank);
56
57     for(Edge edge: edges)
58     {
59         int u = edge.u;
60         int v = edge.v;
61
62         if(findSet(u, parent) != findSet(v, parent))
63         {
64             UnionSet(u, v, parent, rank);
65             mstEdges.push_back(edge);
66             totalWeight += edge.weight;
67         }
68     }
69
70     cout<<"Edge Weight<<endl;
71     for(Edge edge: mstEdges)
72     {
73         cout<<edge.u<<" - "<<edge.v<<" " <<edge.weight<<endl;
74     }
75     cout<<totalWeight<<endl;
76 }
77
78 int main()
79 {
80     vector<vector<int>> graph = {
81         // provided sparse graph
82         {0, 1, 2, 0, 0, 0},
83         {1, 0, 3, 0, 0, 0},
84         {2, 3, 0, 4, 5, 6},
85         {0, 0, 4, 0, 7, 0},
86         {0, 0, 5, 7, 0, 8},
87         {0, 0, 6, 0, 8, 0},
88     };
89
90     auto Start = chrono::high_resolution_clock::now();
91     kruskal(graph.size(), graph);
92     auto End = chrono::high_resolution_clock::now();
93     chrono::duration<double, milli> duration = End - Start;
94
95     cout<<duration.count() <<" ms"<<endl;
96
97     return 0;
98 }
```

```
test_kruskal.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X
55 makeSet(V, parent, rank);
56
57 for(Edge edge: edges)
58 {
59     int u = edge.u;
60     int v = edge.v;
61     if(findSet(u, parent) != findSet(v, parent))
62     {
63         UnionSet(u, v, parent, rank);
64         mstEdges.push_back(edge);
65         totalWeight += edge.weight;
66     }
67 }
68
69 cout<<"Edge Weight<<endl;
70 for(Edge edge: mstEdges)
71 {
72     cout<<edge.u<<" - "<<edge.v<<" " <<edge.weight<<endl;
73 }
74 cout<<totalWeight<<endl;
75
76 }
77
78 int main()
79 {
80     vector<vector<int>> graph = {
81         // provided sparse graph
82         {0, 1, 2, 0, 0, 0},
83         {1, 0, 3, 0, 0, 0},
84         {2, 3, 0, 4, 5, 6},
85         {0, 0, 4, 0, 7, 0},
86         {0, 0, 5, 7, 0, 8},
87         {0, 0, 6, 0, 8, 0},
88     };
89
90     auto Start = chrono::high_resolution_clock::now();
91     kruskal(graph.size(), graph);
92     auto End = chrono::high_resolution_clock::now();
93     chrono::duration<double, milli> duration = End - Start;
94
95     cout<<duration.count() <<" ms"<<endl;
96
97     return 0;
98 }
```

# Prim's Algorithm

```
test_primes.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  void prim(vector<vector<int>>& graph, int start=0)
6  {
7      int V = graph.size();
8      vector<int> key(V, INT_MAX);
9      vector<int> parent(V, -1);
10     vector<bool> inMst(V, false);
11     vector<pair<int,int>> mstEdges;
12     priority_queue<pair<int,int>, vector<pair<int,int>>, greater<pair<int,int>>> minHeap;
13     int totalWeight = 0;
14
15     key[start] = 0;
16     minHeap.push({0, start});
17
18     while(!minHeap.empty())
19     {
20         int u = minHeap.top().second;
21         int weight = minHeap.top().first;
22         minHeap.pop();
23
24         if(inMst[u]) continue;
25
26         inMst[u] = true;
27         totalWeight += weight;
28
29         if(parent[u] != -1)
30             mstEdges.push_back({parent[u], u});
31
32         for(int v=0; v<V; v++)
33         {
34             if(!inMst[v] && graph[u][v]!=0&&graph[u][v]<key[v])
35             {
36                 key[v] = graph[u][v];
37
38                 for(int v=0; v<V; v++)
39                 {
40                     if(!inMst[v] && graph[u][v]!=0&&graph[u][v]<key[v])
41                     {
42                         key[v] = graph[u][v];
43                         parent[v] = u;
44                         minHeap.push({graph[u][v], v});
45                     }
46                 }
47             }
48         }
49     }
50
51     cout<<"Edges   Weight"<<endl;
52     for(auto edge: mstEdges)
53     {
54         cout<<edge.first<<" - "<<edge.second<<"
55         cout<<edge.first<<" - "<<edge.second<<"
56     }
57     cout<<"Total weight:"<<totalWeight<<endl;
58
59     Process returned 0 (0x0)   execution time : 0.080 s
60     Press any key to continue.
```

```
test_primes.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

test_kruskal.cpp X test_primes.cpp X

37     parent[v] = u;
38     minHeap.push({graph[u][v], v});
39 }
40 }
41 }
42
43 cout<<"Edges   Weight"<<endl;
44 for(auto edge: mstEdges)
45 {
46     cout<<edge.first<<" - "<<edge.second<<"   "<<graph[edge.first][edge.second]<<endl;
47 }
48 cout<<"Total weight:"<<totalWeight<<endl;
49
50 }
51
52 int main()
53 {
54     vector<vector<int>> graph = {
55         // provided sparse graph
56         {0, 1, 2, 0, 0, 0},
57         {1, 0, 3, 0, 0, 0},
58         {2, 3, 0, 4, 5, 6},
59         {0, 0, 4, 0, 7, 0},
60         {0, 0, 5, 7, 0, 8},
61         {0, 0, 6, 0, 8, 0},
62     };
63     auto Start = chrono::high_resolution_clock::now();
64     prim(graph);
65     auto End = chrono::high_resolution_clock::now();
66     chrono::duration<double, milli> duration = End - Start;
67     cout<<duration.count()<<" ms"<<endl;
68     return 0;
69 }

31     for(int v=0; v<V; v++)
32     {
33         if(!inMst[v] && graph[u][v]!=0&&graph[u][v]<key[v])
34         {
35             key[v] = graph[u][v];
36             parent[v] = u;
37             minHeap.push({graph[u][v], v});
38         }
39     }
40 }
41
42 cout<<"Edges   Weight"<<endl;
43 for(auto edge: mstEdges)
44 {
45     cout<<edge.first<<" - "<<edge.second<<"
46 }
47 cout<<"Total weight:"<<totalWeight<<endl;
48
49 Process returned 0 (0x0)   execution time : 0.080 s
50 Press any key to continue.
```