

## Task-1: Graphs Graph A and Graph B – apply Prim's algorithms to find the Minimum Spanning Tree (MST).

```
Prims.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here x kru.cpp x Prims.cpp x
1 #include<bits/stdc++.h>
2 #define pii pair<int, int>
3
4 using namespace std;
5 void prims(unordered_map<int, vector<pii>> &adj, int n, int st)
6 {
7     vector<int> key(n, INT_MAX);
8     vector<int> parrent(n, -1);
9     vector<bool> mst(n, false);
10    priority_queue<pii, vector<pii>, greater<pii>> pq;
11    key[st]=0;
12    pq.push({key[st], st});
13    while (!pq.empty())
14    {
15        int u = pq.top().second;
16        pq.pop();
17        if(mst[u]) continue;
18        mst[u] = true;
19        for(auto &edge: adj[u])
20        {
21            int v = edge.first;
22            int wt = edge.second;
23            if (!mst[v] and wt < key[v])
24            {
25                key[v]=wt;
26                parrent[v]=u;
27                pq.push({key[v], v});
28            }
29        }
30    }
31    int mcost=0;
```

G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default 66° Search ENG INTL 12:15 AM 12/30/2024

```
Prims.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here x kru.cpp x Prims.cpp x
32 char Ar[] = {'A','B','C','D','E','F','G','H','I','J'};
33 for(int i=0; i<n; i++){
34     if(parrent[i] != -1){
35         mcost += key[i];
36         cout<< Ar[parrent[i]] << " - " << Ar[i] << " (Weight: " << key[i] << ")\n";
37     }
38 }
39 cout<<"Minimum cost= " << mcost << endl;
40 }
41
42 int main(){
43     int n=6;
44     unordered_map<int, vector<pair<int, int>>> adj;
45     adj[0].push_back({1, 1}); // A-B
46     adj[0].push_back({2, 2}); // A-C
47     adj[1].push_back({0, 1}); // B-A
48     adj[1].push_back({2, 3}); // B-C
49     adj[2].push_back({0, 2}); // C-A
50     adj[2].push_back({1, 3}); // C-B
51     adj[2].push_back({3, 4}); // C-D
52     adj[2].push_back({4, 5}); // C-E
53     adj[2].push_back({5, 6}); // C-F
54     adj[3].push_back({2, 4}); // D-C
55     adj[3].push_back({4, 7}); // D-E
56     adj[4].push_back({2, 4}); // E-C
57     adj[4].push_back({3, 7}); // E-D
58     adj[4].push_back({5, 8}); // E-F
59     adj[5].push_back({2, 6}); // F-C
60     adj[5].push_back({4, 8}); // F-E
61     adj[9].push_back({6, 8});
62     cout<<"For Graph A: " << endl;
```

G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert 66° Search 12:15 AM 12/30/2024

Prims.cpp - Code::Blocks 20.03

```
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
```

<global>

Start here x kru.cpp x Prims.cpp x

```
68 cout<<"\nFor Graph B:"<<endl;
69 adj.clear();
70 n=10;
71 adj[0].push_back({1,3}); adj[0].push_back({3,4});
72 adj[0].push_back({4,4}); adj[1].push_back({4,2});
73 adj[1].push_back({5,3}); adj[1].push_back({2,10});
74 adj[1].push_back({0,3}); adj[2].push_back({5,6});
75 adj[2].push_back({6,1}); adj[2].push_back({1,10});
76 adj[3].push_back({7,6}); adj[3].push_back({4,5});
77 adj[3].push_back({0,4}); adj[4].push_back({7,2});
78 adj[4].push_back({8,1}); adj[4].push_back({5,11});
79 adj[4].push_back({0,8}); adj[4].push_back({1,2});
80 adj[4].push_back({3,5}); adj[5].push_back({6,2});
81 adj[5].push_back({8,3}); adj[5].push_back({9,11});
82 adj[5].push_back({1,3}); adj[5].push_back({4,11});
83 adj[5].push_back({2,6}); adj[6].push_back({9,8});
84 adj[6].push_back({2,1}); adj[6].push_back({5,2});
85 adj[7].push_back({3,6}); adj[7].push_back({4,2});
86 adj[7].push_back({8,4}); adj[8].push_back({4,1});
87 adj[8].push_back({5,3}); adj[8].push_back({9,7});
88 adj[9].push_back({8,7}); adj[9].push_back({5,11});
89 adj[9].push_back({6,8});
90
91 start = clock();
92 prims(adj, n, 0);
93 end = clock();
94 executionTime = (double(end-start)/CLOCKS_PER_SEC)*1000;
95 cout<<"Execution Time: "<<executionTime<<"ms"<<endl;
96 return 0;
97
98
```

"G:\3rd year 1st semester\P x

For Graph A:  
A - B (Weight: 1)  
A - C (Weight: 2)  
C - D (Weight: 4)  
C - E (Weight: 5)  
C - F (Weight: 6)  
Minimum cost= 18  
Execution Time: 35ns

For Graph B:  
A - B (Weight: 3)  
G - C (Weight: 1)  
A - D (Weight: 4)  
B - E (Weight: 2)  
B - F (Weight: 3)  
F - G (Weight: 2)  
E - H (Weight: 2)  
E - I (Weight: 1)  
I - J (Weight: 7)  
Minimum cost= 25  
Execution Time: 11ns

Process returned 0 (0x0) execution time : 0.333

G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/W

## Task-2: Graphs Graph A and Graph B – apply Kruskal's algorithms to find the Minimum Spanning Tree (MST).

kru.cpp - Code::Blocks 20.03

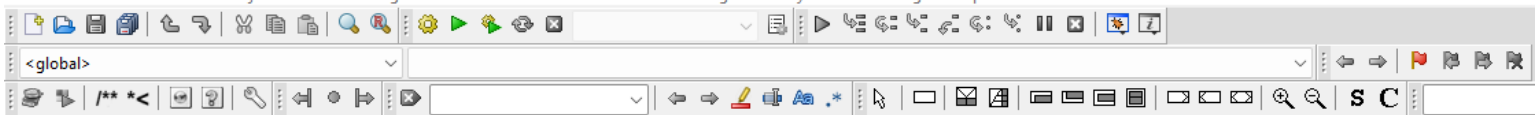
```
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
```

<global>

Start here x kru.cpp x Prims.cpp x

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 bool cmp(vector<int> &a, vector<int> &b)
4 {
5     return a[2]<b[2];
6 }
7 void makeSet(vector<int> &parent, vector<int> &rank, int n){
8     for( int i=0; i<n; i++){
9         rank[i]=0;
10        parent[i]=i;
11    }
12 }
13 int findParent(vector<int> &parent, int node){
14     if(parent[node]==node)
15         return node;
16     return parent[node]=findParent(parent, parent[node]);
17 }
18 void unionSet(int u, int v, vector<int> &parent, vector<int> &rank){
19     u = findParent(parent, u);
20     v = findParent(parent, v);
21     if(rank[u]<rank[v])
22         parent[u]=v;
23     else if(rank[v]<rank[u])
24         parent[v]=u;
25     else {
26         parent[v]=u;
27         rank[u]++;
28     }
29 }
30 int main()
31 {
```

G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default

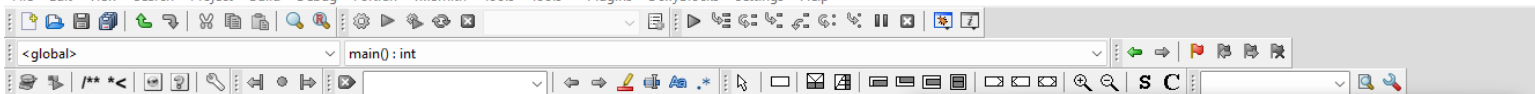


```

32  int n = 10;
33  char Ar[] = {'A','B','C','D','E','F','G','H','I','J'};
34  cout<<"For Graph A:"<<endl;
35  vector<vector<int>> edges = {
36      {0,1,3},{0,3,4},{0,4,4},{1,4,2},{1,5,3},{1,2,10},{2,5,6},{2,6,1},
37      {3,7,6},{3,4,5},{4,7,2},{4,8,1},{4,5,11},{5,6,2},{5,8,3},{5,9,11},{6,9,8},{7,8,4},{8,9,7}
38  };
39  sort(edges.begin(), edges.end(), cmp);
40  vector<int> parent(n);
41  vector<int> rank(n);
42  makeSet(parent, rank, n);
43  int minWeight = 0;
44  clock_t start = clock();
45  for(int i=0; i<edges.size(); i++){
46      int u=findParent(parent, edges[i][0]);
47      int v=findParent(parent, edges[i][1]);
48      int wt = edges[i][2];
49      if(u!=v){
50          cout<<Ar[edges[i][0]]<< " -> " <<Ar[edges[i][1]] <<" : " << edges[i][2]<<endl;
51          minWeight += wt;
52          unionSet(u, v, parent, rank);
53      }
54  }
55  clock_t end = clock();
56  double executionTime = (double(end-start)/CLOCKS_PER_SEC)*1000;
57  cout<<"Minimum Cost: " <<minWeight<<endl;
58  cout<<"Execution Time: " <<executionTime<<"ns"<<endl<<endl;
59
60  n = 6;
61  char Ar1[] = {'A','B','C','D','E','F','G','H','I','J'};
62  cout<<"For Graph B:"<<endl;

```

G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Re



```

64  {0, 1, 1},
65  {0, 2, 2},
66  {1, 2, 3},
67  {2, 3, 4},
68  {2, 4, 5},
69  {2, 5, 6},
70  {3, 4, 7},
71  {4, 5, 8}
72  };
73  sort(edges.begin(), edges.end(), cmp);
74  vector<int> Parent(n);
75  vector<int> Rank(n);
76  makeSet(Parent, Rank, n);
77  minWeight = 0;
78  start = clock();
79  for(int i=0; i<edges.size(); i++){
80      int u=findParent(Parent, edges[i][0]);
81      int v=findParent(Parent, edges[i][1]);
82      int wt = edges[i][2];
83      if(u!=v){
84          cout<<Ar1[edges[i][0]]<< " -> " <<Ar1[edges[i][1]] <<" : " << edges[i][2]<<endl;
85          minWeight += wt;
86          unionSet(u, v, Parent, Rank);
87      }
88  }
89  end = clock();
90  executionTime = (double(end-start)/CLOCKS_PER_SEC)*1000;
91  cout<<"Minimum Cost: " <<minWeight<<endl;
92  cout<<"Execution Time: " <<executionTime<<"ns"<<endl;
93  return 0;
94

```

"G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++

For Graph B:

C -> G : 1  
E -> I : 1  
B -> E : 2  
F -> G : 2  
E -> H : 2  
B -> F : 3  
A -> B : 3  
A -> D : 4  
I -> J : 7  
Minimum Cost: 25  
Execution Time: 30ns

For Graph A:

A -> B : 1  
A -> C : 2  
C -> D : 4  
C -> E : 5  
C -> F : 6  
Minimum Cost: 18  
Execution Time: 9ns

Process returned 0 (0x0) execution time : 0.235 s  
Press any key to continue.

G:\3rd year 1st semester\Practical Lab\Algorithm Design and Analysis (sessional)\after ... C/C++ Windows (CR+LF) WINDOWS-1252 Line 37, Col 24, Pos 976 Insert Read/Write default

