

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

<global>
main(): void

1 // BE2005F062 AKASH SHRIDHARAN
2 #include<stdio.h>
3
4 #define MAX 30
5
6 typedef struct edge
7 {
8     int u,v,w;
9 }edge;
10
11 typedef struct edgelist
12 {
13     edge data[MAX];
14     int n;
15 }edgelist;
16
17 edgelist elist;
18
19 int G[MAX][MAX],n;
20 edgelist spanlist;
21
22 void kruskal();
23 int find(int belongs[],int vertexno);
24 void union1(int belongs[],int c1,int c2);
25 void sort();
26 void print();
27
28 void kruskal()
29 {
30     int belongs[MAX],i,j,cno1,cno2;
31     elist.n=0;
32
33     for(i=1;i<n;i++)
34         for(j=0;j<i;j++)
35             if(G[i][j]!=0)
```

```
C:\Users\akash\Desktop\5th_sem_books&PPTs\DAA\DAA-lab-works\KRUSKALS_ALGO.exe
Enter number of vertices:3
Enter the adjacency matrix:
9
8
7
6
5
4
3
2
1
0 1 2
0 0 3

Cost of the spanning tree=5
Process returned 0 (0x0) execution time : 66.834 s
Press any key to continue.
```

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

<global>
main(): void

28 void kruskal()
29 {
30     int belongs[MAX], i, j, cno1, cno2;
31     elist.n=0;
32
33     for(i=1; i<n; i++)
34         for(j=0; j<i; j++)
35         {
36             if(G[i][j]!=0)
37             {
38                 elist.data[elist.n].u=i;
39                 elist.data[elist.n].v=j;
40                 elist.data[elist.n].w=G[i][j];
41                 elist.n++;
42             }
43         }
44
45     sort();
46
47     for(i=0; i<n; i++)
48         belongs[i]=i;
49
50     spanlist.n=0;
51
52     for(i=0; i<elist.n; i++)
53     {
54         cno1=find(belongs, elist.data[i].u);
55         cno2=find(belongs, elist.data[i].v);
56
57         if(cno1!=cno2)
58         {
59             spanlist.data[spanlist.n]=elist.data[i];
60             spanlist.n=spanlist.n+1;
61             union1(belongs, cno1, cno2);
62         }
63     }
}
```

```
C:\Users\akash\Desktop\5th_sem_books&PPTs\DAA\DAA-lab-works\KRUSKALS_ALGO.exe
Enter number of vertices:3
Enter the adjacency matrix:
9
8
7
6
5
4
3
2
3
2      1      2
2      0      3

Cost of the spanning tree=5
Process returned 0 (0x0)   execution time : 66.834 s
Press any key to continue.
```

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

<global>
main(): void

61 union1(belongs, cno1, cno2);
62 }
63 }
64 }
65 int find(int belongs[], int vertexno)
66 {
67     return(belongs[vertexno]);
68 }
69
70 void union1(int belongs[], int c1, int c2)
71 {
72     int i;
73
74     for(i=0; i<n; i++)
75         if(belongs[i]==c2)
76             belongs[i]=c1;
77 }
78 void sort()
79 {
80     int i, j;
81     edge temp;
82
83     for(i=1; i<elist.n; i++)
84         for(j=0; j<elist.n-1; j++)
85             if(elist.data[j].w>elist.data[j+1].w)
86             {
87                 temp=elist.data[j];
88                 elist.data[j]=elist.data[j+1];
89                 elist.data[j+1]=temp;
90             }
91 }
92 void print()
93 {
94     int i, cost=0;
95
96     for(i=0; i<spanlist.n; i++)
```

```
C:\Users\akash\Desktop\5th_sem_books&PPTs\DAA\DAA-lab-works\KRUSKALS_ALGO.exe

Enter number of vertices:3
Enter the adjacency matrix:
9
8
7
6
5
4
3
2
1
0

2 1 2
2 0 3

Cost of the spanning tree=5
Process returned 0 (0x0) execution time : 66.834 s
Press any key to continue.
```

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

<global>
main(): void

91 }
92 void print()
93 {
94     int i, cost=0;
95
96     for(i=0; i<spanlist.n; i++)
97     {
98         printf("\n%d\t%d\t%d", spanlist.data[i].u, spanlist.data[i].v, spanlist.data[i].w);
99         cost=cost+spanlist.data[i].w;
100     }
101
102     printf("\n\nCost of the spanning tree=%d", cost);
103 }
104
105 int main()
106 {
107     int i, j, total_cost;
108
109     printf("\nEnter number of vertices:");
110
111     scanf("%d", &n);
112
113     printf("\nEnter the adjacency matrix:\n");
114
115     for(i=0; i<n; i++)
116         for(j=0; j<n; j++)
117             scanf("%d", &G[i][j]);
118
119     kruskal();
120     print();
121 }
122
123
124
125
```

```
C:\Users\akash\Desktop\5th_sem_books&PPTs\DAA\DAA-lab-works\KRUSKALS_ALGO.exe

Enter number of vertices:3
Enter the adjacency matrix:
9
8
7
6
5
4
3
2
1
0

2      1      2
2      0      3

Cost of the spanning tree=5
Process returned 0 (0x0)   execution time : 66.834 s
Press any key to continue.
```