| Subject: DAA LAB | Software: Ubuntu |
|---|---|
| | Hardware: Corei5 |
| Branch: CSE / Semester: 4th | Page No. 26 / Prog No. 07 |

**PROBLEM STATEMENT:** Write a program to find minimum cost Spanning Tree of a given connected undirected graph using Prim's Algorithm.

**ALGORITHM & CODE:**

```
#include <stdio.h>
#include <limits.h>
#include <stdbool.h>

#define MAX 20

int main()
{ int graph [MAX][MAX];
  int V, weight;
  Printf("Enter number of vertices (max %d):", MAX);
  Scanf("%d", &V);

  for (int i = 0; i < V; i++)
  { for (int j = 0; j < V; j++)
    { graph [i][j] = 0;
    }
  }
Printf ("\n Enter edge weights (Enter 0 if no edge exists):\n");
for (int i = 0; i < V; i++)
{ for (int j = i + 1 ; j < V ; j++)
  {
    Printf ("weight between vertex %d and %d : ", i, j);
    Scanf ("%d", &weight);
```

**INPUT GIVEN**

**OUTPUT OBTAINED**

**REMARKS**

**GRADE:**

Signature of Faculty
Date:

Signature of Student Rudranarayan Sahoo
Date: 03/04/25

| Subject: DAA LAB | Software: Ubuntu |
|---|---|
| | Hardware: Core i5 |
| Branch: CSE / Semester: 4th | Page No. 27 / Prog No. 07 |

PROBLEM STATEMENT

ALGORITHM & CODE:

```
        graph [i][j] = weight;
        graph [j][i] = weight;
    }
}

int parent [V];
int key [V];
bool inMST [V];

for (int i=0; i<V; i++)
{
    key[i] = INT_MAX;
    inMST [i] = false;
}

key [0] = 0;
parent [0] = -1;

for (int count = 0; count < V-1; count ++)
{
    int min_key = INT_MAX, u;
    for (int v = 0; v < V; v++)
    {if (inMST [v] == false && key [v] < min_key)
        {
            min_key = key [v];
            u = v;
        }
    }
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE:

Signature of Faculty
Date:

Signature of Student Rudranarayan Sahoo
Date:

| Subject: DAA LAB | Software: Ubuntu |
|---|---|
| | Hardware: Corce i5 |
| Branch: CSE / Semester: 4th | Page No. 28 / Prog No. 07 |

PROBLEM STATEMENT

ALGORITHM & CODE:

```
inMST [v] = true;
for (int v = 0; v < V; v++)
{ if (graph [u][v] && inMST [v] == false && graph [u][v]
< key [v])
    {
      parent [v] = u;
      key [v] = graph [u][v];
    }
}
}
Printf ("\n Minimum Spanning Tree Edges : \n");
Printf ("Edge \t Weight \n");
int total_weight = 0;
for (int i = 1; i < v; i++)
  {
   Printf ("%d - %d \t %d \n", parent[i], i, graph [i]
[parent [i]]);
   total_weight += graph [i] [parent [i]];
}
Printf ("\n Total weight of MST : %d \n", total_weight );
return 0;
}
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

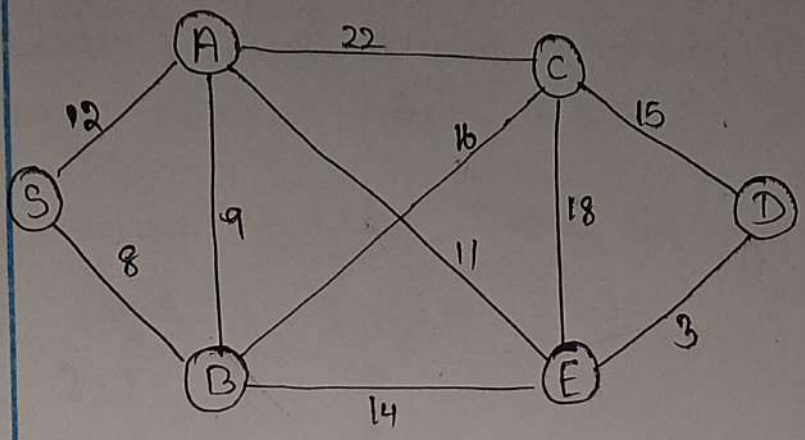
| GRADE: | Signature of Faculty<br>Date: | Signature of Student Rudranarayan Sahoo<br>Date: |
|---|---|---|

| Subject : DAA LAB | Software : Ubuntu |
|---|---|
| | Hardware : Core i5 |
| Branch : CSE / Semester : 4th | Page No. 29 / Prog No. 07 |

PROBLEM STATEMENT

ALGORITHM & CODE :

INPUT




OUTPUT:

```
Enter number of vertices (max 20): 6

Enter edge weights (Enter 0 if no edge exists):
Weight between vertex 0 and 1 : 12
Weight between vertex 0 and 1 : 8
Weight between vertex 0 and 3 : 0
Weight between vertex 0 and 4 : 0
weight between vertex 0 and 5 : 0
weight between vertex 1 and 2 : 9
weight between vertex 1 and 3 : 22
weight between vertex 1 and 4 : 0
weight between vertex 1 and 5 : 11
weight between vertex 2 and 3 : 16
weight between vertex 2 and 4 : 0
```

| INPUT GIVEN | |
|---|---|
| OUTPUT OBTAINED | |
| REMARKS | |
| GRADE : | Signature of Faculty / Date : |
| | Signature of Student Rudranarayan Sahoo / Date : |

Subject: DAA LAB

Software: Ubuntu

Hardware: Corce i5

Branch: CSE | Semester: 4th | Page No. 30 | Prog No. 07

PROBLEM STATEMENT

ALGORITHM & CODE:

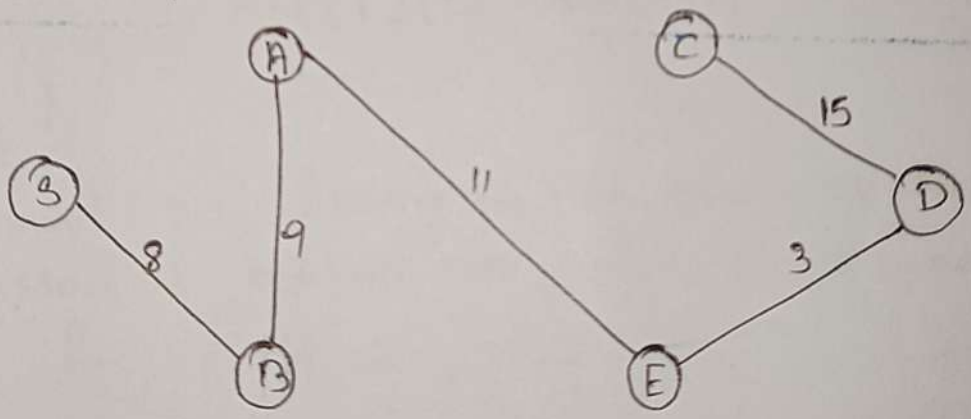
weight between vertex 2 and 5 : 14
weight between vertex 3 and 4 : 15
weight between vertex 3 and 5 : 18
weight between vertex 4 and 5 : 3

Minimum Spanning Tree Edges:

| Edge | Weight |
|---|---|
| 2-1 | 9 |
| 0-2 | 8 |
| 4-3 | 15 |
| 5-4 | 3 |
| 1-5 | 11 |

Total weight of MST : 46

MST Graph:




INPUT GIVEN

OUTPUT OBTAINED

REMARKS