**East West University**
**Department of Computer Science and Engineering**

# Course: CSE 246( Algorithms)
# Section - 02

| **Submitted To :** | **Submitted By :** |
|---|---|
| **Dr. Md. Tauhid Bin Iqbal**<br>Assistant Professor<br>Department of CSE<br>East West University | **Name** : Hafsa Ferdousi<br>**Student ID** : 2023-3-60-321<br>**Department** : CSE<br>**Date** : 15 / 12 / 2025 |

**Kruskal's Algorithm**

```
#include <iostream>
#include <algorithm>
using namespace std;

#define V 5
#define E 7

struct Edge {
   int src, dest, weight;
};

int findParent(int parent[], int i)
{
   if (parent[i] == i)
      return i;
   return findParent(parent, parent[i]);
}

void unionSet(int parent[], int x, int y)
{
   int xset = findParent(parent, x);
   int yset = findParent(parent, y);
   parent[xset] = yset;
}

bool compare(Edge a, Edge b)
{
   return a.weight < b.weight;
}
void kruskalMST(Edge edges[])
{
   Edge result[V];
   int parent[V];

   for (int i = 0; i < V; i++)
      parent[i] = i;

   sort(edges, edges + E, compare);

   int e = 0;
   int i = 0;

   while (e < V - 1)
   {
      Edge next = edges[i++];

      int x = findParent(parent, next.src);
```

```
        int y = findParent(parent, next.dest);

        if (x != y)
        {
            result[e++] = next;
            unionSet(parent, x, y);
        }
    }

    cout << "Edge \tWeight\n";
    for (i = 0; i < e; i++)
        cout << result[i].src << " - " << result[i].dest
             << " \t" << result[i].weight << endl;
}

int main()
{

    Edge edges[E] = {
        {0, 1, 2},
        {0, 2, 3},
        {1, 3, 15},
        {1, 4, 2},
        {2, 3, 7},
        {2, 4, 1},
        {3, 4, 0}
    };

    kruskalMST(edges);
    return 0;
}
```

**OUTPUT:**

```
Edge    Weight
3 - 4   0
2 - 4   1
0 - 1   2
1 - 4   2

Process returned 0 (0x0)   execution time : 0.413 s
Press any key to continue.
```