مقرر خوارزميات و بنى المعطيات2 - جلسة العملي الثالثة

خوارزميات إيجاد شجرة التغطية الصغرى (Minimum Spanning Tree)

خوارزمية بريم لإيجاد شجرة التغطية الصغرى (Prim algorithm):

تطبق الخوارزمية من أجل بيان غير موجه و موزون .

```
/* prim minimum spanning tree from node 1 */
#include <iostream>
#include <String>
using namespace std;
#define  INFF 9999

struct edge{
        int x1;
        int x2;
    };
void prim(int n , int w[][5],edge* F){
        int vnear;
        int min;
        int i,j;
        int f_size=0;
        //the nearest node for each vertex from 2 to n
        int * nearest=new int [n+1];
        //distance of each vertex and the nearest node
        int *distance=new int[n+1];
        for(i=2;i<=n;i++){
            nearest[i]=1;
            distance[i]=w[0][i-1];
        }
        //find vertex with minimum distance
        for(i=0;i<n;i++)
        {
            min =INFF;
            for(j=2;j<=n;j++)
            {

                if(distance[j]<min&&distance[j]>=0)
                {
                    min=distance[j];
                    vnear=j;
                }
```

```
            }
            edge new_edge;
            new_edge.x1=vnear;
            new_edge.x2=nearest[vnear];
            //add edge to MST
            F[f_size]=new_edge;
            f_size++;
            //vnear is visited
            distance[vnear]=-1;

        for(j=2;j<=n;j++){
        if (w[j-1][vnear-1]<distance[j]){
            distance[j]=w[j-1][vnear-1];
            nearest[j]=vnear;
        }
        }
        }
}

int main()
{
    int weigth_matrix[5][5]={{INFF,INFF,3,1,0},
                             {INFF,6,3,0,1},
                             {2,4,0,3,3},
                             {INFF,6,4,0,5},
                             {INFF,INFF,2,5,0}
                             };
     //MST maximum edge number is n-1
     edge * F =new edge [4];
     prim(5,weigth_matrix,F);
     int i;
     for(i=0;i<4;i++)
        cout<<F[i].x1<<","<<F[i].x2<<endl;


}
```