SC165

Problem Statement : You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.

```
#include <iostream>
#include <stdio.h>
using namespace std;

#define ROW 10
#define COL 10
#define infi 9999

   class prims{
      int graph[ROW][COL],nodes;
      int s,e,cost;
      public:
         void createGraph();
         void display();
         void primsAlgo();
   };

   void prims::display()
   {
      for(int i=0;i<nodes;i++){
         for(int j=0;j<nodes;j++){
            cout<<graph[i][j]<<" ";
         }
         cout<<"\n";
      }
   }

   void prims::createGraph(){
      int i,j;
      cout<<"enter total office:";
      cin>>nodes;
      cout<<"\n enter adjacency matrix:\n";
      for(i=0;i<nodes;i++){
         for(j=i;j<nodes;j++){
            cout<<"enter distance between "<<i<<"and"<<j<<endl;
```

```
            cin>>graph[i][j];
            graph[j][i]=graph[i][j];

        }
    }
    for(i=0;i<nodes;i++){
        for(j=0;j<nodes;j++){
            if(graph[i][j]==0)
            graph[i][j]=infi;
        }
    }
}

void prims::primsAlgo(){
    int selected[ROW],i,j,ne=0;
    int zero=0,one=1,min=0,x,y;
    int cost=0;
    for(i=0;i<nodes;i++)
        selected[i]=zero;
        selected[0]=one;

{
    min=infi;
    while(ne<nodes-1){
        min=infi;

        for(i=0;i<nodes;i++){
            if(selected[i]==one){
                for(j=0;j<nodes;j++){
                    if(selected[j]==zero){
                        if (min>graph[i][j]){
                            min=graph[i][j];
                            x=i;
                            y=j;
                        }
                    }

                }
            }
        }
        selected[y]=one;
        cout<<"\n"<<x<<"-->"<<y;
        cost+=graph[x][y];
        ne++;
```

```
}
cout<<"\nTotal cost is:"<<cost<<endl;

}
}

int main(){
prims MST;
cout<<"\nprims algorithm to connect several office\n";
MST.createGraph();
MST.display();
MST.primsAlgo();
}
```

Output

```
/tmp/giXrdiDJKZ.o

prims algorithm to connect several office
enter total office:4

 enter adjacency matrix:
enter distance between 0and0
0
enter distance between 0and1
100
enter distance between 0and2
250
enter distance between 0and3
300
enter distance between 1and1
0
enter distance between 1and2
0
enter distance between 1and3
200
enter distance between 2and2
0
enter distance between 2and3
150
enter distance between 3and3
0
9999 100 250 300
100 9999 9999 200
250 9999 9999 150
300 200 150 9999

0-->1
1-->3
3-->2
Total cost is:450
```