```cpp
//日期: 2018/ 时间:
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <vector>
using namespace std;

const int maxv = 1000;
const int INF = 0x7fffffff;

//邻接矩阵版
//#define p1
//邻接表版
#define p2

#ifdef p1
int n,G[maxv][maxv];
int d[maxv];
bool vis[maxv] = {false};

int prim(){          //默认0号为初始节点，返回最小生成树的边权之和
    fill(d,d+maxv,INF);
    d[0] = 0;
    int ans = 0;

    for(int i=0;i<n;i++){
        int u=-1,MIN = INF;
        for(int j=0;j<n;j++){
            if(vis[j] == false && d[j] < MIN){
                u = j;
                MIN = d[j];
            }
        }

        if(u == -1) return -1;
        vis[u] = true;
        ans += d[u];

        for(int v=0;v<n;v++){        //d[]表示顶点v与集合s的距离
            if(vis[v]==false && G[u][v]!=INF && G[u][v] < d[v]){
                d[v] = G[u][v];
            }
        }
    }
}

#endif

#ifdef p2
struct Node{
    int v,dis;
};
vector<Node> adj[maxv];
int n;
int d[maxv];
bool vis[maxv] = {false};
```

```
57
58  int prim(){
59      fill(d,d+maxv,INF);
60      d[0] = 0;
61      int ans = 0;
62
63      for(int i=0;i<n;i++){
64          int u = -1, MIN = INF;
65          for(int j=0;j<n;j++){
66              if(vis[j] == false && d[j] < MIN){
67                  u = -1;
68                  MIN = d[j];
69              }
70          }
71          if(u == -1) return -1;
72          vis[u] = true;
73          ans += d[u];
74
75          for(int j=0;j<adj[u].size();j++){
76              int v = adj[u][j].v;
77              if(vis[v] == false && adj[u][j].dis < d[v]){
78                  d[v] = adj[u][v].dis;
79              }
80          }
81
82      }
83  }
84
85
86  #endif
87
88  int main(){
89
90
91      return 0;
92  }
93
94
```