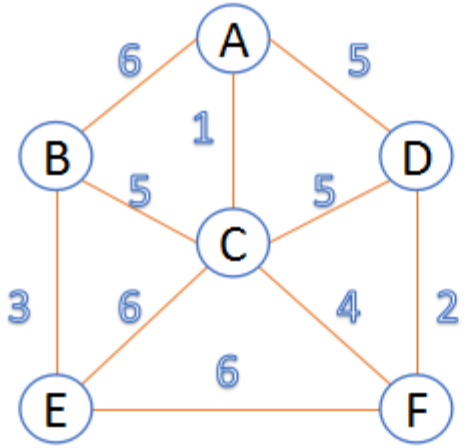
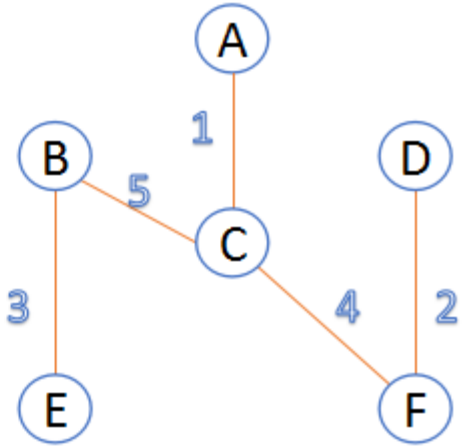
4. 图的最小生成树



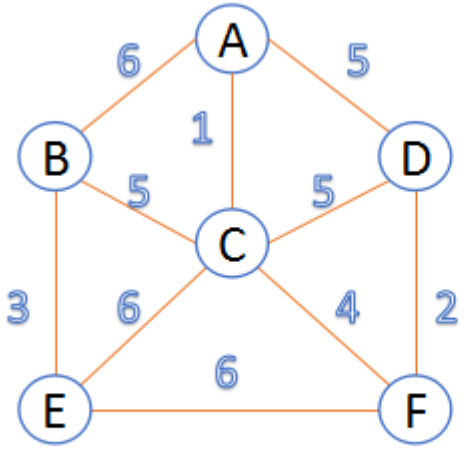
连通网G



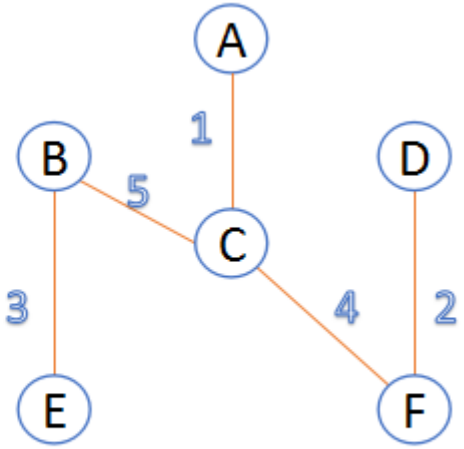
最小生成树

生成树：一个连通图的生成树是指一个连通子图，它含有图中全部n个顶点，但只有足以构成一棵树的n-1条边。一颗有n个顶点的生成树有且仅有n-1条边，如果生成树中再添加一条边，则必定成环。

最小生成树：在连通网的所有生成树中，所有边的代价和最小的生成树，称为最小生成树。



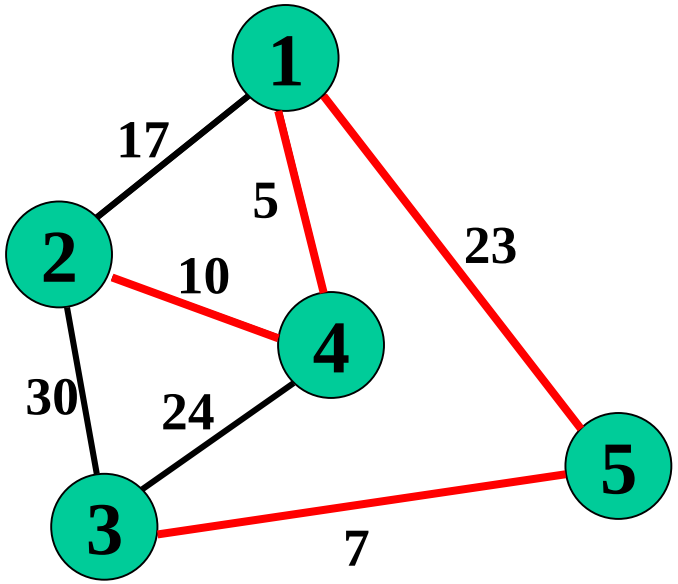
连通网G



最小生成树

最小生成树：

含有n个结点的连通图，从中选n-1条边，保持n个点中任意两点是连通的，并且n-1条边的权值和最小。这n个点和这n-1条边就成为原图的最小生成树。



最小生成树算法：

- 克鲁斯卡尔（kruskal）
- 普里姆算法（prim）

◆克鲁斯卡尔（kruskal）

算法步骤：

1、把图中的边按权值w[i]从小到大排序。

2、按从小到大的顺序依次向树中加边。

在添加每一条边（u，v）时，如果u和V两个点都已在树中，一旦添加，就回构成回路，所以放弃该边，在向后找下一条边。

3、直到添加n-1条边。

特点：给定图的边。

关键：加边时不能构成回路：边的两个顶点是否已在树中。

定义边结构体：

```
struct  Edge{int u,v,w;};
Edge  e[10001];
读入边m条边；
把边e以w为关键字从小到大排序；
int   kru()
        cnt=0;树的边数
        ans=0;树边的长度和
        for(int i=0;i<m;i++){
                对于每条边e[i];
                查找顶点e[i].u的集合x;
                查找顶点e[i].v的集合y;
                if(x!=y){
                        合并x和y；
                        cnt++;
                        ans+=e[i].w;
                        if(cnt==n-1) 返回ans;
                }
    }
```

5

7

1 2 17

1 4 5

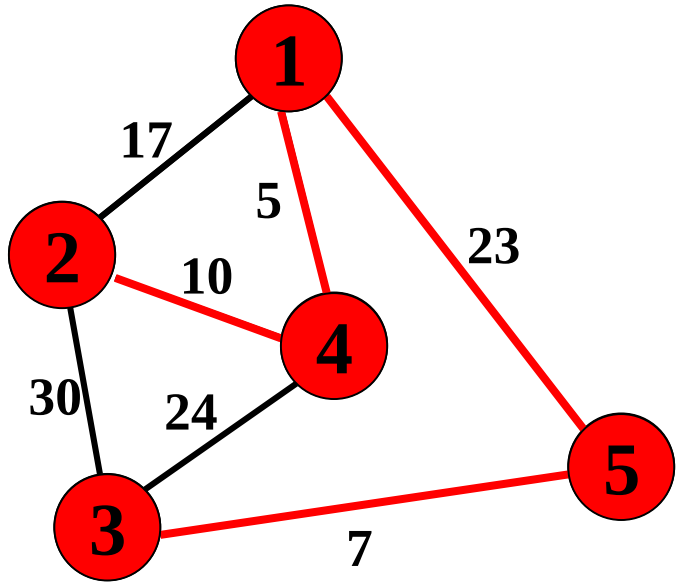
1 5 23

2 3 30

2 4 10

3 4 24

3 5 7



1. int cmp(int i,int j){return w[i]<w[j];}

2. int find(int x){return p[x]==x?x:p[x]=find(p[x]);}

```
const int maxe=100100;
const int maxn=10010;
using namespace std;
struct Edge{
    int u,v,w;
};
Edge e[maxe];
int f[maxn];
int n,m;
int cmp(Edge a,Edge b){return a.w<b.w;}
int Find(int x){return f[x]==0?x:f[x]=Find(f[x]);}
```

```
17 int kruskal(){
18     int cnt=0;
19     long long ans=0;
20     for(int i=0;i<m;i++){
21         int x=Find(e[i].u);
22         int y=Find(e[i].v);
23         if(x!=y){
24             f[x]=y;
25             ans+=e[i].w;
26             if(++cnt>=n-1) return ans;
27         }
28     }
29 }
```

【训练题目】

1391 局域网（net）
1348 城市公交网建设问题
1392 繁忙的都市(city)
1393 联络员(liaison)

◆普里姆算法（prim）

任意结点开始（不妨设为v1）构造最小生成树：

首先把这个结点包括进生成树里，然后在那些其一个端点已在生成树里、另一端点还未在生成树里的所有边中找出权最小的一条边，并把这条边、包括不在生成树的另一端点包括进生成树，…。依次类推，直至将所有结点都包括进生成树为止。

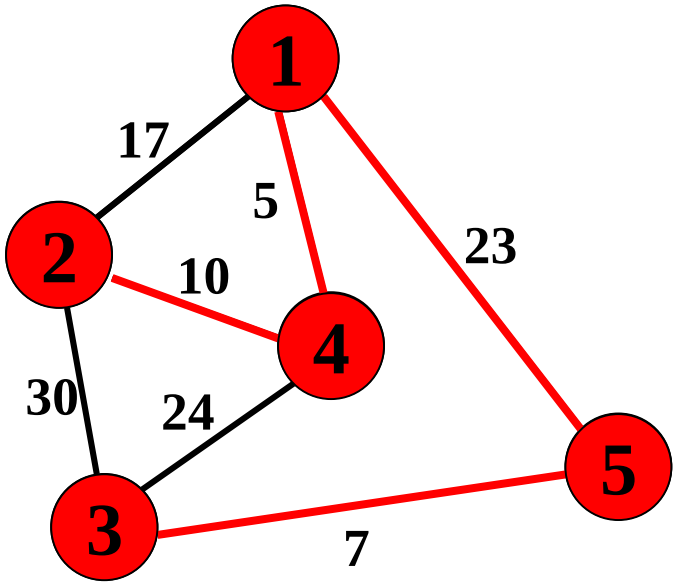
算法描述：

step 1,将生成树赋为空。任选一点放进生成树里。

step 2,在那些其一个端点已在生成树里、另一端点还未在生成树里的所有边中找出权最小的一条边，并把这条边以及不在生成树的另一端点包括进生成树。

step 3,重复step 2，直至将所有结点都包括进生成树为止。

图的存储方式：邻接矩阵或邻接表


1
2
3
4
5
17
5
23
10
30
24
7

```cpp
int prim(){
    memset(d,0x7f,sizeof(d));
    memset(vis,0,sizeof(vis));
    for(int i=1;i<=n;i++)d[i]=a[1][i];
    d[1]=0;
    vis[1]=1;
    int ans=0;
    for(int i=1;i<n;i++){
        int k=0;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&d[j]<d[k]) k=j;
        vis[k]=1;//if k=0
        ans+=d[k];
        for(int j=1;j<=n;j++)
            if(!vis[j]) d[j]=min(d[j],a[k][j]);
    }
    return ans;
}
```

1350:最短网络 Agri-Net