# 最小树形图

## Minimum Directed Spanning Tree

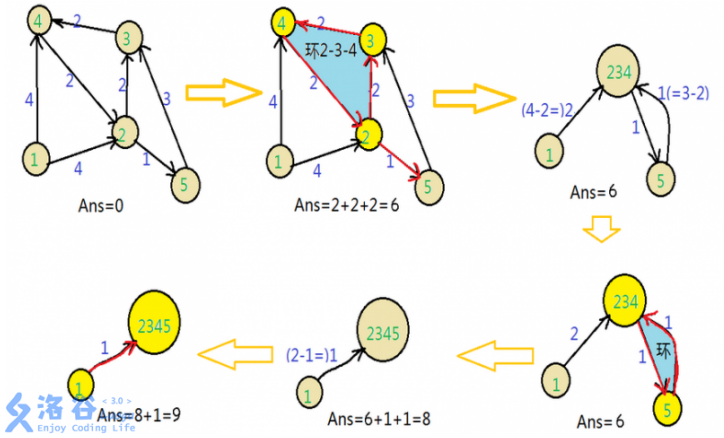
把一棵树的所有边定向为父节点到子节点，构成的有向图称为树形图。

最小树形图即一个图能生成的所有树形图中权值最小者。

## 朱-刘算法 Chu-Liu / Edmonds' Algorithm

Idea：缩点+贪心。朱-刘算法能解决给定根的最小树形图。

- 找出所有除根节点以外其它点的最小入边，如果有一个非根节点没有入边，那么不存在树形图；
- 如果这些入边没有构成环，那么它们就构成了最小树形图；
- 如果有环，则缩点，并更改这个环的入边的权值：设 $v$ 是环内一点，在环内指向 $v$ 的边权为 $w$，$(v',v)$ 是环外指向 $v$ 的一条边，则置 $w(v',v) := w(v',v) - w$. 原因是选了这条入边相当于删去环中 $v$ 的入边。对新图反复执行上述步骤。




Comlexity：$O(VE)$

Extension：不固定根节点：建一个虚拟节点，向所有点连接 $\sum w+1$（即非常大）的边，以虚拟节点为根做最小树形图。如果最小树形图权值 $\geqslant 2\sum w+2$，说明选了两条虚边，那么原图没有最小树形图；否则减去 $\sum w+1$ 就是原图的最小树形图边权和。

Code：

```
int n, m, rt;

struct Edge{
    int u, v, w; // from u to v
}edge[M];
int pre[N], ine[N], vis[N], id[N];
int Chu_Liu(){
    int res = 0;
    int tot = n;
    while(1){
        for(int i = 1; i <= tot; i++)   ine[i] = INF;
        for(int i = 1; i <= m; i++)
            if(edge[i].u != edge[i].v && ine[edge[i].v] > edge[i].w)
                ine[edge[i].v] = edge[i].w, pre[edge[i].v] = edge[i].u;
        for(int i = 1; i <= tot; i++) // check if no solution
            if(i != rt && ine[i] == INF)
                return -1;
        int cnt = 0;
        for(int i = 1; i <= tot; i++)   vis[i] = id[i] = 0;
        for(int i = 1; i <= tot; i++){
            if(i == rt) continue;
            res += ine[i];
            int now = i;
            while(vis[now] != i && !id[now] && now != rt) // find loops
                vis[now] = i, now = pre[now];
```

```
            if(vis[now] == i){ // find a loop, mark it
                id[now] = ++cnt;
                for(int p = pre[now]; p != now; p = pre[p])
                    id[p] = cnt;
            }
        }
        if(cnt == 0)    return res; // no loop -> find a solution
        for(int i = 1; i <= tot; i++)
            if(!id[i])  id[i] = ++cnt;
        for(int i = 1; i <= m; i++){ // rebuild the graph
            if(id[edge[i].u] != id[edge[i].v])  edge[i].w -= ine[edge[i].v];
            edge[i].u = id[edge[i].u], edge[i].v = id[edge[i].v];
        }
        rt = id[rt];
        tot = cnt;
    }
}
```