

最小树形图

Minimum Directed Spanning Tree

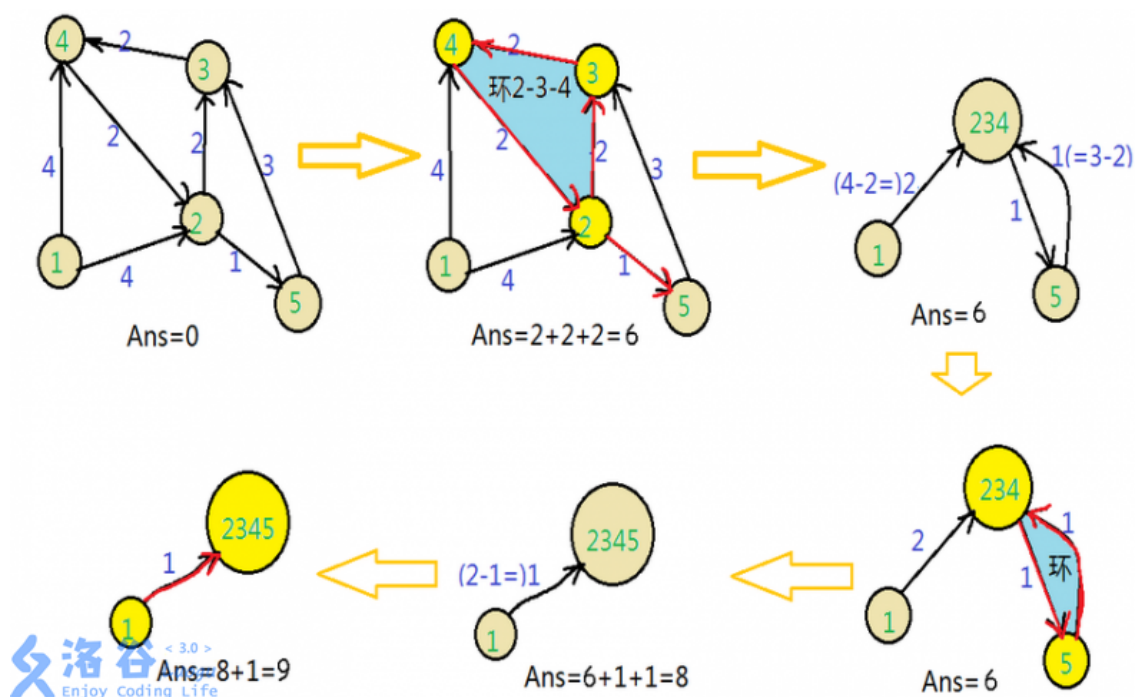
把一棵树的所有边定向为父节点到子节点，构成的有向图称为树形图。

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。



Complexity: $O(VE)$

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 $\geq 2 \sum w + 2$ ，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:

```
1  int n, m, rt;
2
3  struct Edge{
4      int u, v, w; // from u to v
5  }edge[M];
6  int pre[N], ine[N], vis[N], id[N];
7  int Chu_Liu(){
8      int res = 0;
9      int tot = n;
10     while(1){
11         for(int i = 1; i <= tot; i++)    ine[i] = INF;
12         for(int i = 1; i <= m; i++)
13             if(edge[i].u != edge[i].v && ine[edge[i].v] > edge[i].w)
14                 ine[edge[i].v] = edge[i].w, pre[edge[i].v] =
edge[i].u;
15         for(int i = 1; i <= tot; i++) // check if no solution
16             if(i != rt && ine[i] == INF)
17                 return -1;
18         int cnt = 0;
19         for(int i = 1; i <= tot; i++)    vis[i] = id[i] = 0;
20         for(int i = 1; i <= tot; i++){
21             if(i == rt) continue;
22             res += ine[i];
23             int now = i;
24             while(vis[now] != i && !id[now] && now != rt) // find
loops
25                 vis[now] = i, now = pre[now];
26             if(vis[now] == i){ // find a loop, mark it
27                 id[now] = ++cnt;
28                 for(int p = pre[now]; p != now; p = pre[p])
29                     id[p] = cnt;
30             }
31         }
32         if(cnt == 0)    return res; // no loop -> find a solution
33         for(int i = 1; i <= tot; i++)
34             if(!id[i])    id[i] = ++cnt;
35         for(int i = 1; i <= m; i++){ // rebuild the graph
36             if(id[edge[i].u] != id[edge[i].v])    edge[i].w -=
ine[edge[i].v];
37             edge[i].u = id[edge[i].u], edge[i].v = id[edge[i].v];
38         }
```

```
39 |         rt = id[rt];  
40 |         tot = cnt;  
41 |     }  
42 | }
```