

GGBeng

编程，首先要多敲，其次才是学！

博客园 新随笔 管理

随笔- 551 文章- 0 评论- 21

昵称: GGBeng
园龄: 2年3个月
粉丝: 166
关注: 9
[+加关注](#)

最小树形图——朱刘算法

一、相关定义

定义：设 $G = (V, E)$ 是一个有向图，它具有下述性质：

1. G 中不包含有向环；
2. 存在一个顶点 v_i ，它不是任何弧的终点，而 V 中的其它顶点都恰好是唯一的一条弧的终点，则称 G 是以 v_i 为根的树形图。

最小树形图就是有向图 $G = (V, E)$ 中以 v_i 为根的树形图中权值和最小的那一个。

另一种说法：最小树形图，就是给有向带权图一个特殊的点root，求一棵以root为根节点的树使得该树的总权值最小。

性质：最小树形图基于贪心和缩点的思想。

缩点：将几个点看成一个点，所有连到这几个点的边都视为连到收缩点，所有从这几个点连出的边都视为从收缩点连出

二、算法描述

【概述】

为了求一个图的最小树形图，①先求出最短弧集合 E_0 ；②如果 E_0 不存在，则图的最小树形图也不存在；③如果 E_0 存在且不具有环，则 E_0 就是最小树形图；④如果 E_0 存在但是存在有向环，则把这个环收缩成一个点 u ，形成新的图 G_1 ，然后对 G_1 继续求其的最小树形图，直到求到图 G_i ，如果 G_i 不具有最小树形图，那么此图不存在最小树形图，如果 G_i 存在最小树形图，那么逐层展开，就得到了原图的最小树形图。

【实现细节】

设根结点为 v_0 ，

- (1) 求最短弧集合 E_0

从所有以 $v_i (i \neq 0)$ 为终点的弧中取一条最短的，若对于点 i ，没有入边，则不存在最小树形图，算法结束；如果能取，则得到由 n 个点和 $n-1$ 条边组成的图 G 的一个子图 G' ，这个子图的权值一定是最小的，但是不一定是一棵树。

- (2) 检查 E_0

若 E_0 没有有向环且不包含收缩点，则计算结束， E_0 就是图 G 以 v_0 为根的最小树形图；若 E_0 含有有向环，则转入步骤(3)；若 E_0 没有有向环，但是存在收缩点，转到步骤(4)。

- (3) 收缩 G 中的有向环

把 G 中的环 C 收缩成点 u ，对于图 G 中两端都属于 C 的边就会被收缩掉，其他弧仍然保留，得到一个新的图 G_1 ， G_1 中以收缩点为终点的弧的长度要变化。变化的规则是：设点 v 在环 C 中，且环中指向 v 的边的权值为 w ，点 v' 不在环 C 中，则对于 G 中的每一条边 $\langle v', v \rangle$ ，在 G_1 中有边 $\langle v', u \rangle$ 和其对应，且权值 $W_{G_1}(\langle v', u \rangle) = W_G(\langle v', v \rangle) - w$ ；对于图 G 中以环 C 中的

< 2019年11月 >						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

搜索

积分与排名

积分 - 113814
排名 - 4827

随笔分类

C++Primer习题(20)
C++学习(81)
Effective C++(5)
Java学习之路(15)
Linux入门(32)
Windows编程——MFC方式(13)
笔试题(18)
递归(3)
动态规划(1)
二叉树(8)
计算机网络(6)
计算机知识基础(6)
每天一笔(28)
排序(6)
散列表(1)
数据结构(16)
数据结构（基础篇）(20)
数据库学习(1)
数据库语言-SQL(16)
思想聚焦(21)
图算法(1)
网络编程(7)
系统编程进阶(2)
系统编程入门(15)
系统编程习题(3)

阅读排行榜

1. C++中substr函数的用法(62770)
2. C++STL——优先队列(38298)
3. 数据结构4——并查集（入门）(24623)
4. C++STL——队列(13656)

点为起点的边 $\langle v', v \rangle$ ，在图 G_1 中有边 $\langle u, v' \rangle$ ，则 $W_{G_1}(\langle u, v' \rangle) = W_G(\langle v', v \rangle)$ 。有一点需要注意，在这里生成的图 G_1 可能存在重边。

对于图 G 和 G_1 ：

①如果图 G_1 中没有以 v_0 为根的最小树形图，则图 G 也没有；

②如果 G_1 中有一 v_0 为根的最小树形图，则可按照步骤（4）的展开方法得到图 G 的最小树形图。

所以，应该对于图 G_1 代到(1)中反复求其最小树形图，直到 G_1 的最小树形图 u 求出。

• （4）展开收缩点

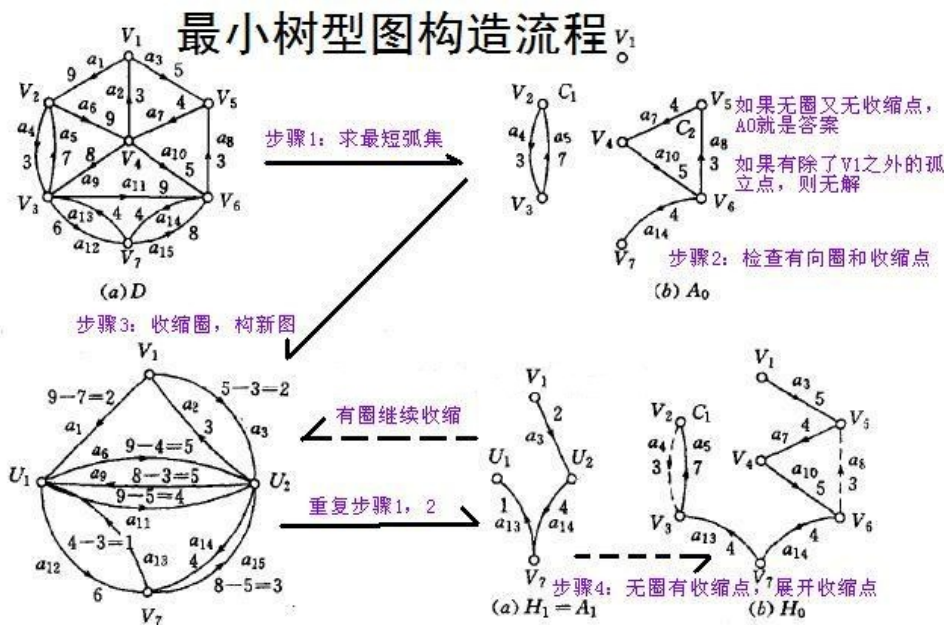
假设图 G_1 的最小树形图为 T_1 ，那么 T_1 中所有的弧都属于图 G 的最小树形图 T 。将 G_1 的一个收缩点 u 展开成环 C ，从 C 中去掉与 T_1 具有相同终点的弧，其他弧都属于 T 。

【小结】

对最小树形图做个小小的总结：

- 1：清除自环，自环是不可能存在于任何最小树形图中的；
- 2：求出每个顶点的的最小入边；
- 3：判断该图是否存在最小树形图，由 1 可以判定，或者以图中顶点 v 作为根节点遍历该图就能判断是否存在最小树形图；
- 4：找环，之后建立新图，缩点后重新标记。

【图示——最小树形图构造流程】



解读：第一幅图为原始图 G ，首先对于图 G 求其最短弧集合 E_0 ，即第二幅图 G_1 ；然后检查 E_0 是满足条件，在这里，可以看到 G_1 具有两个环，那么把这两个环收缩，如第三幅图所示， U_1 、 U_2 分别为收缩后的点，然后将对应的权值进行更新，起点是环中的点，终点是环外的点，则权值不变。反之，起点是环外的点，终点是环内的点，则权值应该减去 E_0 中指向环内点的权值，形成新的图，如第三幅图，对于其反复求最小树形图，直到不存在最小树形图，或者求得收缩点后的图的最小树形图，然后展开就好了，如第六幅图。

如果只要求计算权值的话，则不需要展开，所有环中权值的和加上其他各个点与点之间，或者收缩点和点之间的权值就是总的权值。

5. 制作一个简易计算器——基于Android Studio实现(11451)
6. 递归从入门到精通(11305)
7. 【转】C++后台开发之我见(8652)
8. 最短路径——SPFA算法(8310)
9. 并查集（进阶）(4995)
10. 最小树形图——朱刘算法(4734)

三、沙场练兵

题目: [hdu 2121 Ice cream's world II](#)

代码:

```

1  #include<iostream>
2  using namespace std;
3  #include<cstdio>
4  #include<cstring>
5  #define MAXN 1005
6  #define INF 0x7f7f7f7f
7  typedef __int64 type;
8  struct node//边的权和顶点
9  {
10     int u, v;
11     type w;
12 }edge[MAXN * MAXN];
13 int pre[MAXN], id[MAXN], vis[MAXN], n, m, pos;
14 type in[MAXN];//存最小入边权,pre[v]为该边的起点
15 type Directed_MST(int root, int V, int E)
16 {
17     type ret = 0;//存最小树形图总权值
18     while(true)
19     {
20         int i;
21         //1.找每个节点的最小入边
22         for( i = 0; i < V; i++)
23             in[i] = INF;//初始化为无穷大
24         for( i = 0; i < E; i++)//遍历每条边
25         {
26             int u = edge[i].u;
27             int v = edge[i].v;
28             if(edge[i].w < in[v] && u != v)//说明顶点v有条权值较小的入边 记录之
29             {
30                 pre[v] = u;//节点u指向v
31                 in[v] = edge[i].w;//最小入边
32                 if(u == root)//这个点就是实际的起点
33                     pos = i;
34             }
35         }
36         for( i = 0; i < V; i++)//判断是否存在最小树形图
37         {
38             if(i == root)
39                 continue;
40             if(in[i] == INF)
41                 return -1;//除了根以外有点没有入边,则根无法到达它 说明它是独立的点 一定不能构成树并
42         }
43         //2.找环
44         int cnt = 0;//记录环数
45         memset(id, -1, sizeof(id));
46         memset(vis, -1, sizeof(vis));
47         in[root] = 0;
48         for( i = 0; i < V; i++) //标记每个环
49         {
50             ret += in[i];//记录权值
51             int v = i;
52             while(vis[v] != i && id[v] == -1 && v != root)
53             {
54                 vis[v] = i;
55                 v = pre[v];
56             }
57             if(v != root && id[v] == -1)
58             {
59                 for(int u = pre[v]; u != v; u = pre[u])
60                     id[u] = cnt;

```

```

61         id[v] = cnt++;
62     }
63 }
64 if(cnt == 0)
65     break; //无环 则break
66 for( i = 0; i < V; i++)
67     if(id[i] == -1)
68         id[i] = cnt++;
69 //3.建立新图 缩点,重新标记
70 for( i = 0; i < E; i++)
71 {
72     int u = edge[i].u;
73     int v = edge[i].v;
74     edge[i].u = id[u];
75     edge[i].v = id[v];
76     if(id[u] != id[v])
77         edge[i].w -= in[v];
78 }
79 V = cnt;
80 root = id[root];
81 }
82 return ret;
83 }
84 int main()
85 {
86     int i;
87     while(scanf("%d%d", &n, &m) != EOF)
88     {
89         type sum = 0;
90         for( i = 0; i < m; i++)
91         {
92             scanf("%d%d%I64d", &edge[i].u, &edge[i].v, &edge[i].w);
93             edge[i].u++; edge[i].v++;
94             sum += edge[i].w;
95         }
96         sum ++;
97         for( i = m; i < m + n; i++)//增加超级节点0,节点0到其余各个节点的边权相同 (此题中 边权要大于0)
98         {
99             edge[i].u = 0;
100             edge[i].v = i - m + 1;
101             edge[i].w = sum;
102         }
103         type ans = Directed_MST(0, n + 1, m + n);
104         //n+1为总结点数,m+n为总边数
105         //ans代表以超级节点0为根的最小树形图的总权值,
106         //将ans减去sum,如果差值小于sum,说明节点0的出度只有1,说明原图是连通图
107         //如果差值>=sum,那么说明节点0的出度不止为1,说明原图不是连通图
108         if(ans == -1 || ans - sum >= sum)
109             puts("impossible");
110         else
111             printf("%I64d %d\n", ans - sum, pos - m);
112         puts("");
113     }
114     return 0;
115 }

```

好文要顶

关注我

收藏该文



GGBeng

关注 - 9

粉丝 - 166

+加关注

1

0

« 上一篇: [边结点结构体的定义及使用](#)» 下一篇: [C++中substr函数的用法](#)

评论

1楼 2019-08-16 18:47 | UniverseofHK

请问一下新增的超级源点为什么是0呀？普通节点的编号不是0~n-1吗？超级源点和普通节点公用0这个编号？

支持(0) 反对(0)

2楼 2019-08-16 18:51 | UniverseofHK

哦，对不起，我懂了！

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

- 【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【活动】京东云服务器_云主机低于1折，低价高性能产品备战双11
- 【培训】马士兵老师强势回归！Java线下课程全免费，双十一大促！
- 【推荐】天翼云双十一翼降到底，云主机11.11元起，抽奖送大礼
- 【推荐】流程自动化专家UiBot，体系化教程成就高薪RPA工程师
- 【福利】个推四大热门移动开发SDK全部免费用一年，限时抢！
- 【推荐】阿里云双11冰点钜惠，热门产品低至一折等你来抢！

相关博文：

- [最小树形图——朱刘算法](#)
 - [最小树形图\(模板\)](#)
 - [最小树形图\(朱刘算法\)](#)
 - [POJ 3164 Command Network （最小树形图 朱刘算法）](#)
 - [POJ 3164 Command Network 最小树形图 朱刘算法](#)
- » [更多推荐...](#)

最新 IT 新闻：

- [白宫警告重返月球计划需要更多资金](#)
 - [Pixel 4让人大失所望 谷歌犯了这么几个错误](#)
 - [三星内存生产设备污染发生在器兴工厂 专家：损失远超10亿韩元](#)
 - [专业软件的强制订阅让开源替代更有吸引力](#)
 - [尼安德特人可能死于现代人类带来的疾病](#)
- » [更多新闻...](#)