

bin3学艺录

抗战二十年

最小树形图

2007-10-29 14:46

学了不少算法，终于第一次学会了以中国人的名字命名的算法，最小树形图的朱-刘算法。前几天看过书、ppt和网上相关的blog，在收缩环关键的那块总看得稀里糊涂。昨晚终于在威士忌的blog上看到了一幅最小树形图的构造流程图，登时大彻大悟。有时一张恰当好处的图表的确胜过万千文字解说。早上起来把它实现了，调试到现在，进一步把各种模糊的细节处理弄清楚了，也终于AC了POJ3164。呵呵

网上关于算法的描述有一些，我总结成bin3版。

有固定根的最小树形图求法O(VE):

首先消除自环，显然自环不在最小树形图中。然后判定是否存在最小树形图，以根为起点DFS一遍即可。

之后进行以下步骤。

设cost为最小树形图总权值。

0.置cost=0。

1.求最短弧集合Ao（一条弧就是一条有向边）

除源点外，为所有其他节点Vi，找到一条以Vi为终点的边，把它加入到集合Ao中。

（加边的方法：所有点到Vi的边中权值最小的边即为该加入的边，记prev[vi]为该边的起点，mincost[vi]为该边的权值）

2.检查Ao中的边是否会形成有向圈，有则到步骤3，无则到步骤4。

（判断方法：利用prev数组，枚举为检查过的点作为搜索的起点，做类似DFS的操作）

3.将有向环缩成一个点。

假设环中的点有（Vk1, Vk2, ..., Vki）总共i个，用缩成的点叫Vk替代，则在压缩后的图中，其他所有不在环中点v到Vk的距离定义如下：

$gh[v][Vk] = \min \{ gh[v][Vkj] - mincost[Vkj] \} \quad (1 \leq j \leq i)$ 而Vk到v的距离为

$gh[Vk][v] = \min \{ gh[Vkj][v] \} \quad (1 \leq j \leq i)$

同时注意更新prev[v]的值，即if(prev[v]==Vkj) prev[v]=Vk

另外cost=cost+mincost[Vkj] (1<=j<=i)

到步骤1.

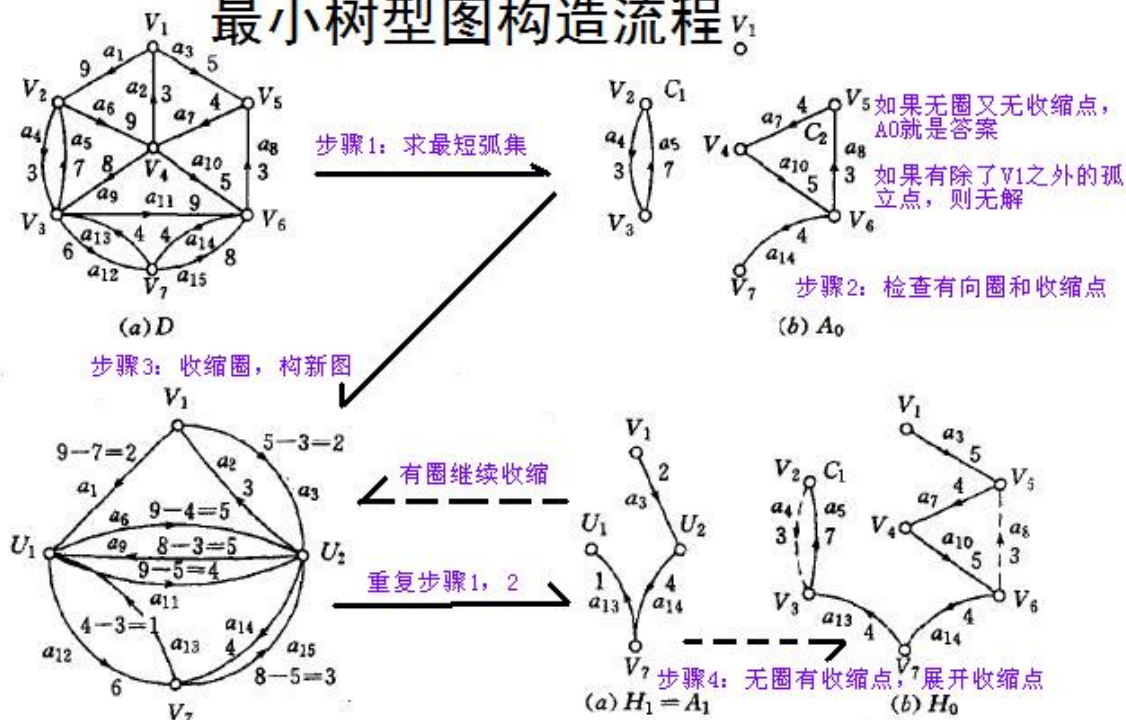
4.cost加上Ao的权值和即为最小树形图总权值。

如要输出最小树形图较烦，没实现过。

找环O(V)，收缩O(E)，总复杂度O(VE)。

那幅对我有莫大帮助的流程图如下，

最小树型图构造流程_{v1}



对于不固定根的最小树形图，wy教主有一巧妙方法。摘录如下：

新加一个点，和每个点连权相同的边，这个权大于原图所有边权的和，这样这个图固定跟的最小树形图和原图不固定跟的最小树形图就是对应的了。

#算法&&数据结构

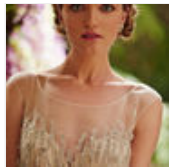
分享到：

举报 浏览(1679) 评论(13) 转载

你可能也喜欢



Naomi Watts 《I nStyle》英国版2



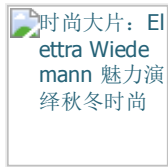
BHLDN 2015春季花园主题婚纱



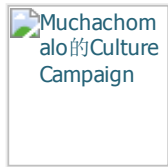
“最佳表演”《W》杂志2015年2月号



【EDITORIAL】两超模Robin P 和 Chris S联手拍摄Billionaire杂志时尚大片



时尚大片：Elettra Wiedemann 魅力演绎秋冬时尚



Muchachomalo的Culture Campaign



Vanessa Mooney 2014秋冬系列珠

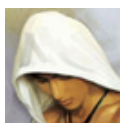
本文最近访客



968_968
3月20日



PigPigZhu
1月15日



daiker0330
10月31日



449280317
10月22日



mathalbus
8月21日



CA72
8月21日



Oyk
7月25日



叁拾叁燎川
7月21日

评论

发布

 哥哥看点我           328

2013-08-03 01:27

[回复](#)

 nudeman 

2012-11-18 19:40

[回复](#)

 limingyao668 LZ的文章看明白了~还真是 有时一张恰当好处的图表的确胜过万千文字解说。

2011-09-09 16:28

[回复](#)

 bin183 回复sciencesuneast: 呵呵

2010-07-03 19:41

[回复](#)

 sciencesuneast 好棒...终于看懂lz的文章了...thx...

2010-06-28 11:02

[回复](#)

 sciencesuneast 晕,看了好多资料了...为什么我就是理解不了

2010-06-28 09:26

[回复](#)

 Moun_ElemeNt 我觉得没有图也可以理解 你讲的很不错。

2010-05-10 14:00

[回复](#)

 mory 嗯,强


2008-07-30 11:14

[回复](#)

 bin183 当然可能产生新环, 每处理完步骤3, 应跳到步骤1(我原来错写成2了)继续求最短弧集合, 直到最短弧集合没环为止呵呵, 不客气


2008-06-10 18:14

[回复](#)

 lnkm 感谢指教! 我再琢磨琢磨。在处理环的过程中是否会产生新环。例如上图: a15的权改为6。那么8-5就变成6-5, 按我理解显然步骤四a图的v1->U2就要改成v7->U2。这样就产生了新环(不知对不对)。如果是这样, 那我的程序就可能没有再处理新环。再次感谢赐教!


2008-06-10 14:14

[回复](#)

 bin183 re lnkm:1、判断回路的函数没问题, 但你那样使用它会对一个环中的m个点进行m次环的处理, 不知你怎么处理的, 应该是没有必要的, 对一个环中所有点统一处理一次就可以的2、估计你的理解有些问题, 找到环后是要把这个环缩为一新点, 环内所有边都将被去除。处理的方向是把环外点指向环内任意点的边指向新点, 环内任意点指向环外点的边变为新点指向相应的环外点, 边权值的计算参照文中的步骤3。参考着图看应该会很快理解的

2008-06-10 10:53

[回复](#)

 lnkm 处理环我这样理解, 不知有没有错: 找到一条从外部指向环的弧, 使得与环内的一条边权值之差最小, 并把环内的该边去除。

2008-06-10 07:25

[回复](#)



lnkm 这道题过了sample和自己的测试数据，提交总是WA。请问怎样判断是否存在回路？请问多次调用这个函数判断可以吗bool HaveCirc(int prevex[],int &k,int n){ bool visited[maxsize]; memset(visited,0,sizeof(visited)); while(!visited[k] && k!=0) //逆着边的方向遍历，遍历到一个遍历过的结点，说明存在回路。遍历到根节点（k=0）说明当前结点不在环上。 { visited[k]=true; k=prevex[k]; } if(k==0) //访问到顶点说明该顶点不在环上 return false; else return true;}比如： for(t=1;t

2008-06-09 22:44

[回复](#)

[帮助中心](#) | [空间客服](#) | [投诉中心](#) | [空间协议](#)

©2015 Baidu