

# 浅谈动态树的相关问题及简单拓展



长沙市雅礼中学 黄志翱

April 26, 2014

# 论文概述

全文分成四个部分：

1. 什么是动态树问题。
2. 如何使用Link/cut tree维护路径信息。
3. dfs序，欧拉遍历树在维护子树信息方面的应用。
4. 动态树问题的一些简单拓展。

# 什么是动态树

动态树是对一类问题的称呼。  
很多时候专指LCT。

# 动态树问题

1. 添加一条边或者删除一条边，维护树的形态。
2. 树上的路径操作和查询。
3. 子树操作和查询。

# 为何要研究动态树问题

很多问题都往往包含了动态树问题作为其子问题。

LCT简单，灵活，方便拓展。

随着时代在进步，OI中出现了一些有意思的问题和解法。

有意思的问题

BZOJ 3153

BZOJ 3466

...

# 树的形态固定

1. 树链剖分，可以解决静态的路径和子树维护问题。
2. 点分治，常用于解决路径查询和统计问题。

# 树的形态不固定



# 树的形态不固定

点分治

动态维护点分治!

WC 2014 flower

# 树的形态不固定

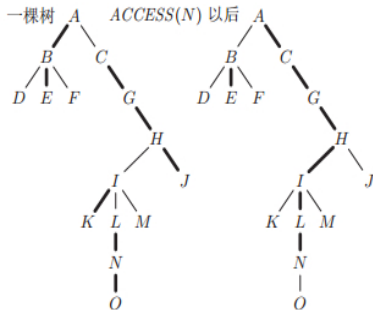
树链剖分

动态维护树链剖分!

Link/cut tree

# Link/cut tree

splay维护实边的树链剖分。



# Link/cut tree

access操作，取出一个点到根的链。

splay的翻转操作，实现换根。

时间复杂度为均摊 $O(\log n)$

# 如何使用LCT

1. 每次可以添加一条边，或者删除一条边。
2. 给一条链上每个点加上一个数。
3. 询问一条链的权值和。

能够用平衡树维护的序列问题，都有可能通过LCT推广到树上。

# LCT的简单应用

BZOJ 3091: 维护路径的最大子段和。

IOI 2011 elephant

# LCT解决简单图论问题

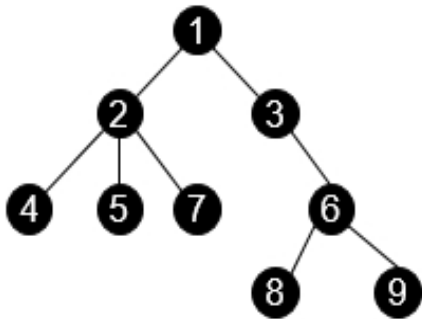
1. 只有加边的最小生成树
2. 最小极差路径
3. 动态维护图是否是二分图

# 如何维护子树信息

LCT是对链的剖分，并非天生支持。  
相应的工具：dfs序和ETT。



## dfs序和ETT



dfs序: 1 2 4 5 7 3 6 8 9

欧拉遍历序: 1 2 4 2 5 2 7 2 1 3 6 8 6 9 6 3 1

## dfs序和ETT

子树和区间一一对应，使用平衡树维护。

## dfs序和ETT

子树和区间一一对应，使用平衡树维护。

### 简单应用

- 动态维护子树大小
- 可持久化dfs序
- 动态维护图的连通性。

# 回顾

- 动态树问题
- Link/cut tree
- dfs序和ETT

# 简单拓展

LCT是一种强大的数据结构!

## 子树信息维护：回到LCT

LCT可以统计子树信息的。



一个点的子树分为实边和虚边两个部分。

维护每个点连出的虚边的信息和，在实链中进行统计。

## 例：查询树上的最远点

给定一棵树，要求支持加边和删边，查询离某个点距离最远的点。

## 例：查询树上的最远点

给定一棵树，要求支持加边和删边，查询离某个点距离最远的点。

每条实链维护离链的端点最远的点。

维护虚边的最远点的优先队列。



# LCT与链翻转

求一个子树中的数的和。要求支持将某条路径上的权值进行翻转。

权值与树的形态分离

LCT中维护虚边的信息和树的形态。

对于每条链维护权值的线段树。

LCT还可以做一些更“有意思”的事情。

接下来的两个问题都是通过对LCT进行简单拓展可以得到的。

# sone1

sxyz里有一群神犇。

要求同时维护子树信息和路径信息，并支持路径修改和子树修改。

## 第一种做法：ETT+LCT

ETT中从一个顶点开始的连续的一段为原树中的路径。

改变ETT的遍历顺序，使得一个点第一个儿子对应于LCT中的实边。

时间复杂度： $O(n \log^2 n)$ 。

## 第二种做法：Self adjusting top tree

名字听起来很可怕。

## 第二种做法：修改版LCT

这样听起来好多了。

## 维护哪些信息

维护哪些信息：

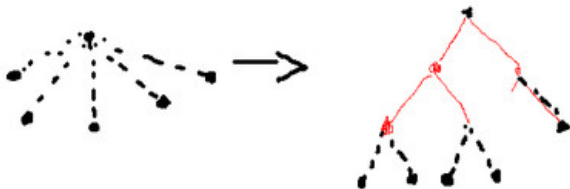
1. 对于每条链统计链内部的点的信息。
2. 所有从这条链上的点连出的虚边对应的子树的信息的和（不包括这条链）。
3. 以上两个信息的和，也就是这条链最顶端的点所对应的子树的信息。

对于链上的信息直接使用splay。  
虚边的信息？



# AAA树

其实不叫这名字。



添加虚节点，将虚边组织成二叉树。可以看成一棵平衡树。

# 时间复杂度?

$O(n \log n)$

有大小为96的常数。

## 另一个问题：动态仙人掌

定义仙人掌图为每条边最多属于一个简单环的图。

动态仙人掌问题是要求动态地维护一个仙人掌图。

定义一条路径为两点之间的最短路。

- 动态仙人掌I：加边和删边，查询最短路径的长度。
- 动态仙人掌II：在I的基础上每条边有一个固定的权值 $B$ ，要求查询一条路径上最小的 $B$ 值。
- 动态仙人掌III：在II的基础上要求支持对一段路径的 $B$ 值进行操作。

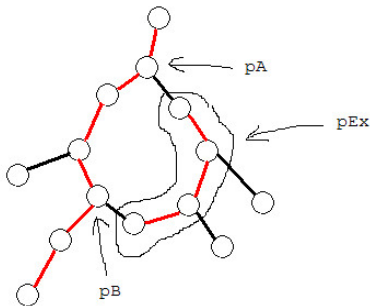
# 动态仙人掌I & II

可以使用LCT维护生成树。

## Link/cut cactus

仿照LCT，为了得到LCC的做法，必须先考虑如何将仙人掌图分成若干条不相交的链。

保证实链为最短路。一个环最多被剖分成两条链。



## Link/cut cactus

维护所有的环及其中的边。

access中的每次操作只会改变一个环对应的两条链。

其余操作仿照LCT。

很明显时间复杂度为 $O(n \log^2 n)$

# Self Adjusting Top Cactus

## Self Adjusting Top Cactus

欢迎有兴趣的人进行更进一步的研究。



# 总结

- 论文从信息维护的角度对动态树问题和解法进行了总结。
- 介绍了LCT的简单拓展。
- 时间有限，很多问题仅仅提及大要，更进一步的细节在论文中会有详细介绍。

# 感谢

感谢廖晓刚老师，朱全民老师和屈运华老师的教导。

感谢国家集训队教练的辛勤付出和CCF提供的机会。

感谢匡正非，刘研绎，毛啸，彭雨翔，杨定澄，刘剑成，谭博文，杨卓林，苏雨峰等同学提供的意见和帮助。

感谢陈立杰，我从他那儿学到了top tree及其它各种各样有趣的算法。

感谢吕凯风在动态仙人掌问题上做出的开创性的研究。

感谢唐翔昊，魏子昆，陈思琪，仇知，徐诗雨，黄骏翔，王子骏等同学陪我度过了快乐的时光。

## 参考文献

- [1] 杨哲,《SPOJ375 QTREE解法的一些研究》,2007年国家集训队作业。
- [2] 陈立杰,《重量平衡树和后缀平衡树在信息学奥赛中的应用》,2013年国家集训队论文。
- [3] 6.851: Advanced Data Structures L20
- [4] 匡正非,黄志翱,《两个冷门图论算法》,2014年信息学奥林匹克冬令营。
- [5] Robert E. Tarjan, Renato F. Werneck, Self-Adjusting Top Trees.
- [6] 吕凯风,《动态仙人掌系列题解》