

LCT总结——应用篇（附题单）（LCT）

为了优化体验（其实是强迫症），蒟蒻把总结拆成了两篇，方便不同学习阶段的Dalao们切换。

LCT总结——概念篇[戳这里](#)

题单

灰常感谢XZY巨佬提供的强力资助！（可参考[XZY巨佬的博客总结](#)）

题单对于系统地学习一个知识点还是有好处的。

所以蒟蒻搜集了各处的LCT题目（其实作为近年新兴的知识点，现有的好题不是很多，有些题树剖也可做）

大概按细化分类进行整理（类比下面的几个细化知识点，会有重复的列举）

同一类中的题目也大概按难度递增吧（太弱了，对每个题的难度定位或许有不准的地方，欢迎讨论！）

维护信息（LCT上的平衡树操作）

- 【Done】洛谷P3690 【模板】Link Cut Tree
- 【Sol.】洛谷P3203 [HNOI2010]弹飞绵羊
- 【Sol.】洛谷P1501 [国家集训队]Tree II
- 【Todo】洛谷P2486 [SDOI2011]染色
- 【Sol.】洛谷P4332 [SHOI2014]三叉神经树

动态维护连通性&双联通分量（可以说是并查集的升级，因为并查集只能连不能断）

- 【Done】洛谷P2147 [SDOI2008]Cave 洞穴勘测
- 【Sol.】洛谷P3950 部落冲突
- 【Sol.】洛谷P2542 [AHOI2005]航线规划
- 【Done】BZOJ4998 星球联盟
- 【Done】BZOJ2959 长跑

维护边权（常用于维护生成树）

- 【Sol.】洛谷P4172 [WC2006]水管局长
- 【Todo】UOJ274温暖会指引我们前行
- 【Sol.】洛谷P4180 [BJWC2010]次小生成树
- 【Sol.】洛谷P4234 最小差值生成树
- 【Sol.】洛谷P2387 [NOI2014]魔法森林

维护子树信息

- 【Done】[COGS2701]动态树
- 【Sol.】洛谷P4219 [BJOI2014]大融合
- 【Sol.】洛谷U19464 山村游历(Wander)
- 【Sol.】洛谷U21715 首都（BZOJ3510权限题qwq）

- [【Todo】洛谷SP2939 QTREE5 - Query on a tree V](#)
- [【Todo】LOJ558「Antileaf's Round」我们的 CPU 遭到攻击](#)

维护树上染色状压DP

- [【Todo】洛谷P2173 \[ZJOI2012\]网络](#)
- [【Sol.】洛谷P3703 \[SDOI2017\]树点涂色](#)
- [【Sol.】洛谷SP16549 QTREE6 - Query on a tree VI](#)
- [【Sol.】洛谷SP16580 QTREE7 - Query on a tree VII](#)
- [【Todo】BZOJ3914 Jabby's shadows](#)

特殊题型

- [【Sol.】洛谷P3613 睡觉困难综合征](#)
- [【Todo】UOJ207 共价大爷游长沙](#)
- [【Sol.】洛谷P3348 \[ZJOI2016\]大森林](#)
- [【Sol.】洛谷P4338 \[ZJOI2018\]历史](#)
- [【Todo】LOJ2289 \[THUWC 2017\]在美妙的数学王国中畅游](#)（数学对于我这种蒟蒻是不可做的TAT）

操作升级！（模板以外的细化总结）

维护链信息（LCT上的平衡树操作）

思路与模板基本都很像了，就不展开讨论了

例题1

[洛谷P1501 \[国家集训队\]Tree II\(点击进入题目\)](#)

感觉难度评定也高了点吧

有了链上乘法和链上加法，那肯定是像线段树区间修改那样使用懒标记了。

放标记的做法来自[\[模板\]线段树2](#)

那里发题解的Dalao们都讲得挺好的，本蒟蒻参考一下。

[蒟蒻的题解在这里](#)

例题2

[洛谷P3203 \[HNOI2010\]弹飞绵羊（点击进入题目）](#)

略带一点思维，但是代码比模板还简单。。。。。

在LCT模板基础上进行灵活运用，可以使代码和程序都做到高效。

[蒟蒻的题解在这里](#)

例题3

[洛谷P4332 \[SHOI2014\]三叉神经树](#)

没有那么裸了，仔细分析并利用好题目的性质

[蒟蒻的题解在这里](#)

动态维护连通性及双连通分量

维护连通性？findroot判一下就好啦！属于模板范畴

至于双连通分量，LCT好像只能资磁加边。。。。。

LCT中的每个点其实代表的是一个双连通分量，维护每个点属于哪个双连通分量需要在外面弄一个并查集维护。

加边时，如果两点不连通就link；

如果连通就说明有环，把环缩成一个点，在并查集里也全部合并在一起

其中要注意很多细节，具体就看下面例题2的代码吧

例题1

[洛谷P3950 部落冲突](#) (点击进入题目)

正解不是LCT。。。。。

只是这里用LCT的思路非常简单甚至于无脑。。。。。

[蒟蒻题解](#)

例题2

[洛谷P2542 \[AHOI2005\]航线规划](#) (点击进入题目)

基础的动态维护双连通分量，具体就看看题解吧

[蒟蒻题解](#)

维护链上的边权信息 (生成树)

普通LCT维护点权，那对于边权呢？比如要获取一条路径上最长的边，等等。

这种维护的最经典的应用，大概就是动态维护一颗最小/大生成树了（需要获取边长度等信息）。我起初有一种想法。

一棵树，除了根节点，每个节点有且仅有一条父边。

那么如果我们只要边的信息，可不可以就把LCT中的点的点权当成父边的边权呢？

在其它不少数据结构中好像是可以这样做的。

只不过LCT性质非常特殊。一旦换了根，原先边的父子关系就破坏了。所以并不能这样做。

那又如何是好呢？

我在网上看到有些博客介绍的方法，是把边置于LCT外，然后在LCT节点中维护父边和重子边的编号，需要更新信息时从外部获取，在access, link, cut时额外更改。

这样好像挺麻烦的，要维护那么多东西。

另一种更抽象的做法，也是我要分享的，就是两个字——拆边，把边视作为一个点，向该边的两个端点连两条边。

只需要维护边权的时候，边权存在LCT中代表边的点权里，因为不需要维护真正的点权，所以LCT中的代表点的点权可以设成空的（0），不会影响信息的正确性。于是就变成了维护点权。

当然了，原先我们的link和cut是针对两个点的，现在有点不一样了（因为两点之间又多了一个代表边的点）

update:发现之前描述的写法有问题，且并不能优化多少常数，在这里更正一下

写法就是link/cut两次，比如

```
link(e[i].id,e[i].x);link(e[i].id,e[i].y);
cut(e[i].id,e[i].x);cut(e[i].id,e[i].y);
```

例题一

[洛谷P4180 \[BJWC2010\]次小生成树](#)

其实这道题用LCT不是正解，不过是可做的（update:我后来又写了LCT），放在这里一下吧。

用LCT维护最小生成树，再去枚举非树边尝试替换树边，更新最优答案。

TPLY巨佬也用LCT做的（某问题导致常数巨大？），[他的题解在这里](#)

[蒟蒻题解在这里](#)

例题二

[洛谷P2387 \[NOI2014\]魔法森林](#)

我太弱啦！这种双关键字的维护我是真的没有思路，边看着题解边打出来的。orz XZY&XZZ两位巨佬，题解点赞！

不过作为LCT维护边权的好题目还是值得分享。

[蒟蒻题解在这里](#)

维护虚子树信息总和与原树信息总和

有时候，我们需要的并不是维护链的信息，而是子树的信息。要知道，LCT是长于维护链的信息，而弱于维护子树信息（这方面不得不承认树剖的用处了，还要赶紧学）。然而动态的连边和断边，又不得不要求我们使用LCT来维护。

其实，我们还是不会束手无策的。

维护子树信息总和的一些常见题型，无非就是询问整棵（原树中的）子树的总大小、权值总和、最值之类的东西。

我们已经可以通过辅助树Splay来获知Splay中的实子树（也就是原树中的一条链）的信息总和。既然原树的边只有实与虚，子树也只有实与虚，那么如果要想维护好原树的信息总和，我们是不是首先要知道虚子树的信息总和？

注：以下设**虚子树信息总和用数组si表示，原树信息总和用s表示**。此处si[x]只包含x所有虚子树

(通过轻边指向x)的信息总和,而s[x]实际上是在LCT中的所有儿子的信息总和(包括辅助树Splay中相对的左右儿子的总和与被轻边所指的绝对的虚子树的总和)。

那么如果我们确定了si[x]的值,是不是就知道了s[x]的值了? 实加虚嘛!

这就是pushup, 代码

```
inline void pushup(int x){
    s[x]=s[c[x][0]]+s[c[x][1]]+si[x]+1;//大小总和, 注意别漏了+1, 自己也要算上
    //或者
    s[x]=s[c[x][0]]+s[c[x][1]]+si[x]+v[x];//权值总和, 自己同样也要算上
}
```

先不对这种方法是怎样来的这个问题追本溯源, 那我们就可以直接考虑si如何维护了。

还是直接假设我们已经提前维护好了si吧。

很显然, si的变化取决于虚实边的变化, 所以现在我们只考虑LCT中的每个操作会对虚实边产生怎样的影响。

- splay(x):显然这一操作只会改变节点在辅助树Splay中的相对位置, 并不会对树中的虚子树信息产生任何影响。过程中的pushup会正确地更新s。
- access(x):这个时候, 我们发现, 循环中每一次splay(x)后, 我们更改了x的右子树, 也就是更改了虚实边, 那么会对信息产生影响。
这里的影响, 无非就是得到了一个虚儿子, 失去了一个虚儿子, 总和没有改变。
于是直接改就好啦, si加上原来右儿子的s, 再减去新的右儿子的s。
与原模板略有不同的代码

```
inline void access(int x){
    for(int y=0;x;x=f[y=x]){
        splay(x);
        si[x]+=s[c[x][1]];
        si[x]-=s[c[x][1]=y];
        pushup(x);
    }
    //如果pushup只是更新原树信息总和s的话, 甚至这里可以不写, 毕竟加一个减一个, 和没变
}
```

- makeroot(x):并无影响。虽然树的形态翻转了, 但是翻转的只是实链。
- findroot(x):显然无影响 (跳左儿子又没改变树的形态)
- split(x,y):显然无影响 (除了调用三个函数外什么都没干)
- link(x,y):这就有影响了。y多了一个轻边指向儿子x, 那么s[y], si[y]都加上s[x]。注意, 多了一个子树, 那么y在LCT中的祖先的s也都要加上s[x]的值, 却没有得到更新。我们在加上s[x]前, 需要额外地把y转到根 (access(y)+splay(y)), y就没有祖先了, 再加, 就不会影响信息的正确性了。
与原模板略有不同的代码

```
//保证连边合法
inline void link(int x,int y){
    split(x,y);//这里不是提取x-y的路径的意思, 是makeroot+access+splay的偷懒写法
    si[f[x]=y]+=s[x];
    pushup(y);
}
//不保证
inline bool link(int x,int y){
    makeroot(x);
    if(findroot(y)==x)return 0;//access+splay已完成
    si[f[x]=y]+=s[x];
    pushup(y);
    return 1;
}
```

- cut(x,y):无影响, 断掉的是一条重边, pushup(y)更新s[y]就好啦。

分析到此完毕。其实好像也就只改了一点点地方。。。。。。不过思路是很巧妙的，值得用心体会。

补充一句，如果要维护子树里的最值，一个套路是在每个节点开一个平衡树维护该节点所有虚子树的最值，以便进行查询和更改。

例题一

洛谷P4219 [BJOI2014]大融合（点击进入题目）

当学会了LCT维护子树信息和以后，这题就变得有些裸了。。。。。。

[蒟蒻题解在这里](#)

例题二

洛谷U19464 山村游历(Wander)（点击进入题目）

注：此题为WC模拟赛试题，版权归出题人Philipsweng所有。小蒟蒻觉得这题出得很不错，于是上传至洛谷个人题库，作为例题与大家分享。数据自测，如有问题欢迎反馈。

题目简述：

没有。。。。。。这是一个阅读题，简述的话就等于告诉你怎么做子

仔细读一下题目吧，这道题水平挺高的。

思路分析：

也有点复杂

然而分析完了以后，又变成裸的了。。。。。。

算了，还是单独建一篇随笔吧。

[蒟蒻题解在这里](#)

维护树上染色联通块

维护方法要视具体情况分析

可能可以开一个LCT然后把同色的点连起来，这样做思路很简单，但实现起来受局限

也可能有多少颜色就开多少个LCT，然后在对应颜色的LCT中连上

升级操作：修改点的颜色

需要转化模型，可参考下面例题2

再次升级操作：修改链为同一种颜色

Link-Cut-Memphis（雾

看一下发明者Memphis巨佬的博客吧

以上两种操作都是在ZJOI2018交流课上听到的Orz

例题一

洛谷P3703 [SDOI2017]树点涂色（点击进入题目）

维护连通块并不麻烦，只是思维难度超大。。。。。。

不会树剖，焉知非福？

[蒟蒻的题解](#)

例题二

洛谷SP16549 QTREE6 - Query on a tree VI（点击进入题目）

一直都把边化为子节点，这次来把点化为父边（雾

[蒟蒻的题解](#)

特殊题型

有一类题目是真的毒瘤，怪异到都几乎认不出它的真面目

似乎并不能把它们归为任何一类LCT题型或者任何一种套路

只好仔细分析题目，发现题目本身的特殊性，再转化为LCT模型解决

然而我什么都分析不出来啊QwQ

例题一

洛谷P3613 睡觉困难综合征

把起床困难综合症——一个按位贪心的题目，完美地套在了LCT中

[蒟蒻题解](#)

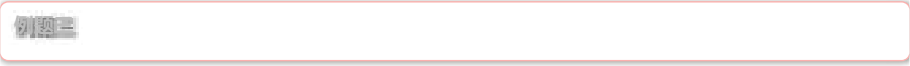
例题二

洛谷P3348 [ZJOI2016]大森林

新技能get：虚点

离线的转移思路也极为巧妙

蒟蒻题解



洛谷P4338 [ZJOI2018]历史

ZJOI唯一可做题TAT

不来一些大力结论，根本做不下去。。。。。

蒟蒻题解

分类：OI——算法总结，数据结构——链剖——LCT

标签：LCT



Flash_Hu

关注 - 78

粉丝 - 152

+加关注

« 上一篇：洛谷CF868F Yet Another Minimization Problem（动态规划，决策单调性，分治）

» 下一篇：洛谷P2900 [USACO08MAR]土地征用Land Acquisition（动态规划，斜率优化，决策单调性，线性规划，单调队列）

posted @ 2018-08-18 20:21 Flash_Hu 阅读(5950) 评论(7) 编辑 收藏

评论列表

#1楼 2018-08-24 11:48 headchen

学习了。mark

#2楼 2018-08-27 19:28 Cgod

FLashHu太太太太太太太太太太太太太太太太太太太太太太太太太太太太辣

#3楼 [楼主] 2018-08-27 19:37 Flash_Hu

@ cx233666
c(x)什么特征多项式都会

#4楼 2018-09-24 21:08 OOOOOOOOOO

flashhu把这两个分开了啊

#5楼 2018-11-25 22:45 bestfy

%%%！感谢博主！

#6楼 2019-03-26 07:47 M_sea

催更“在美妙的数学王国中畅游”

#7楼 [楼主] 2019-03-26 13:35 Flash_Hu

@M_sea
我这个蒟蒻都有半年多没上什么课了，现在导数和泰勒展开什么都不会，还是先退役搞学科算了qaq