

# 自为风月马前卒

## dsu on tree入门

### 目录

- 先瞎扯几句
- Dsu on tree
  - 简介
  - 算法流程
  - 复杂度
  - 模板题
  - 一道比较有意思的题
- 参考资料

[回到顶部](#)

### 先瞎扯几句

说起来我跟这个算法好像还有很深的渊源呢qwq。当时在学业水平考试的考场上，题目都做完了不会做，于是开始xjb出题。突然我想到这么一个题

*attack*有一棵 $n$ 个节点的树，1号节点为根节点，每个点都有一种颜色，他想知道节点 $i$ 的子树中颜色种类数最多的子树的大小是多少。

看起来好像很可做的样子，然而直到考试完我都只想出来一个莫队的暴力。当时我想知道有没有比莫队更优的做法，和zqb讨论了半天也只能搞出一个 $O(n\log^2 n)$ 的平衡树启发式合并

然后！！我就把这题出给校内互测了！！没错，当时是用莫队当的标算！

结果！mjt用一个假的 $O(n)$ 算法<sup>2</sup>过去了因为数据特别水

后来我打算把这题出给另一场比赛，结果到了前一天晚上造数据的时候我

发现不太对，然后把mjt的算法hack了。

去UOJ群里一问才知道这玩意儿是个dsu on tree的sb题。

当时我就这个表情



自己还是太年轻啊%>\_<%

好了好了，来讲算法吧

[回到顶部](#)

## Dsu on tree

### 简介

dsu on tree跟dsu(并查集)是没啥关系，可能是借用了一波启发式合并的思想？

它是用来解决一类树上询问问题，一般这种问题有两个特征

- 1、只有对子树的询问
- 2、没有修改

一般这时候就可以强上dsu on tree了

update：可能特征1不会很显然，就是说题目中不一定明确的问你子树*i*的答案，可能是把问题转化后需要算子树*i*的答案

### 算法流程

考虑暴力怎么写：遍历每个节点—把子树中的所有颜色暴力统计出来更新答案—消除该节点的贡献—继续递归

这肯定是 $O(n^2)$ 的。

dsu on tree巧妙的利用了轻重链剖分的性质，把复杂度降到了 $O(n \log n)$

啥啥啥？你不知道啥叫轻重链剖分？

一句话：对于树上的一个点，与其相连的边中，连向的节点子树大小最大的边叫做重边，其他的边叫轻边

dsu on tree的算法流程是这样的：

对于节点*i*：

- 遍历每一个节点
  - 递归解决所有的轻儿子，同时消除递归产生的影响
- 递归重儿子，不消除递归的影响

- 统计所有轻儿子对答案的影响
- 更新该节点的答案
- 删除所有轻儿子对答案的影响

主体框架长这样

```

1 void dfs(int x, int fa, int opt) {
2     for(all Edge) {
3         if(to == fa) continue;
4         if(to != BigSon[x]) dfs(to, x, 0); //暴力统计轻边的贡献
5     }
6     if(son[x]) dfs(BigSon[x], x, 1); //统计重儿子的贡献，不消除影响
7     add(x); //暴力统计所有轻儿子的贡献
8     ans[x] = NowAns; //更新答案
9     if(!opt) delet(x); //如果需要删除贡献的话就删掉
10 }
11

```

可能你先在会想：为什么都是暴力统计答案？这样复杂度不是 $O(n^2)$ 的么？

那简单的来证一下这东西的复杂度

## 复杂度

性质：一个节点到根的路径上轻边个数不会超过 $\log n$ 条

证明：

设根到该节点有 $x$ 条轻边，该节点的大小为 $y$ ，根据轻重边的定义，轻边所连向的点的不会成为该节点总大小的一般。

这样每经过一条轻边， $y$ 的上限就会 $/2$ ，因此 $y < \frac{n}{2^x}$

因为 $n > 2^x$ ，所以 $x < \log n$

然而这条性质并不能解决问题。

我们考虑一个点会被访问多少次

一个点被访问到，只有两种情况

1、在暴力统计轻边的时候访问到。

根据前面的性质，该次数 $< \log n$

2、通过重边 / 在遍历的时候被访问到

显然只有一次

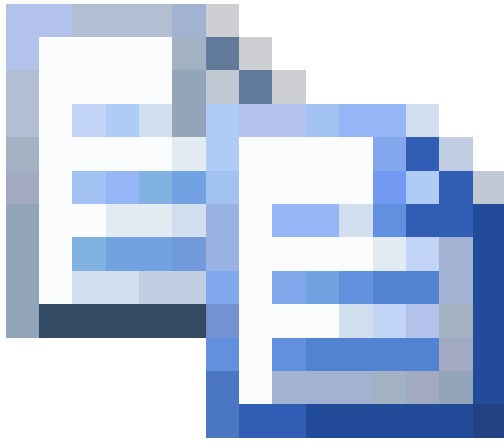
如果统计一个点的贡献的复杂度为 $O(1)$ 的话，该算法的复杂度为 $O(n \log n)$

## 模板题

cf600E. Lomsat gelral

题意：给出一个树，求出每个节点的子树中出现次数最多的颜色的编号和

dsu on tree的模板题，暴力统计即可



```

#include<bits/stdc++.h>
#define LL long long
using namespace std;
const int MAXN = 1e5 + 10;
inline int read() {
    char c = getchar(); int x = 0, f = 1;
    while(c < '0' || c > '9') {if(c == '-') f = -1; c = getchar();}
    while(c >= '0' && c <= '9') x = x * 10 + c - '0', c = getchar();
    return x * f;
}

int N, col[MAXN], son[MAXN], siz[MAXN], cnt[MAXN], Mx, Son;
LL sum = 0, ans[MAXN];
vector<int> v[MAXN];
void dfs(int x, int fa) {
    siz[x] = 1;
    for(int i = 0; i < v[x].size(); i++) {
        int to = v[x][i];
        if(to == fa) continue;
        dfs(to, x);
        siz[x] += siz[to];
        if(siz[to] > siz[son[x]]) son[x] = to; //轻重链剖分
    }
}

void add(int x, int fa, int val) {
    cnt[col[x]] += val; //这里可能会因题目而异
    if(cnt[col[x]] > Mx) Mx = cnt[col[x]], sum = col[x];
    else if(cnt[col[x]] == Mx) sum += (LL)col[x];
    for(int i = 0; i < v[x].size(); i++) {
        int to = v[x][i];
        if(to == fa || to == Son) continue;
        add(to, x, val);
    }
}

void dfs2(int x, int fa, int opt) {
    for(int i = 0; i < v[x].size(); i++) {
        int to = v[x][i];
        if(to == fa) continue;
        if(to != son[x]) dfs2(to, x, 0); //暴力统计轻边的贡献, opt = 0表示递归完成后消除对该点的影响
    }
    if(son[x]) dfs2(son[x], x, 1), Son = son[x]; //统计重儿子的贡献, 不消除影响

    add(x, fa, 1); Son = 0; //暴力统计所有轻儿子的贡献
    ans[x] = sum; //更新答案
    if(!opt) add(x, fa, -1), sum = 0, Mx = 0; //如果需要删除贡献的话就删掉
}

int main() {
    N = read();
    for(int i = 1; i <= N; i++) col[i] = read();
    for(int i = 1; i <= N - 1; i++) {
        int x = read(), y = read();
        v[x].push_back(y); v[y].push_back(x);
    }
    dfs(1, 0);

    dfs2(1, 0, 0);
    for(int i = 1; i <= N; i++) printf("%I64d ", ans[i]);
    return 0;
}

```



## 一道比较有意思的题

不知道老师从哪儿弄的。。。

### 树上统计

(treecnt.c/cpp/pas)

时间限制：1s  
内存限制：256MB

【题目描述】

给定一棵n个点的树。  
定义Tree[L, R]表示为了使得L~R号点两两连通，最少需要选择的边的数量。  
求  $\sum_{L=1}^n \sum_{R=L}^n Tree[L, R]$

【输入格式】

第一行一个数，n表示点数 (n<=100000)  
接下来n-1行每行两个数，x和y，表示一条连接x号点和y号点的边 (x, y<=n)

【输出格式】

输出一个数表示答案

【输入输出样例】

| treecnt.in             | treecnt.out |
|------------------------|-------------|
| 4<br>1 4<br>1 3<br>2 4 | 16          |

【数据范围】

对于20%的数据：n<=10  
对于40%的数据：n<=300  
对于60%的数据：n<=3000  
对于另外20%的数据：树呈一条链  
对于100%的数据：n<=100000

我的题解：<https://www.cnblogs.com/zwfymqz/p/9687296.html>

官方题解：

### 3. 树上统计

### 解题思路

30%: 暴力

40%: 枚举  $L$ ,  $R$  从  $L \sim n$  枚举,  $R$  每增大一个, 更新需要的边 (bfs 实现)

60%: 枚举每条边, 计算每条边的贡献

另外 20% 的数据: 枚举每条边, 计算每条边的贡献

100%:

对于每一条边统计有多少个区间跨过这条边即可

统计这一问题的对偶问题, 有多少个区间没跨过会更方便

使用启发式合并+并查集统计子树内的, 使用启发式合并+set 统计子树外的

[回到顶部](#)

## 参考资料

[Codeforces600E]Lomsat gelral (dsu on the tree)

[trick]dsu on tree

作者: 自为风月马前卒

个人博客<http://attack204.com/>

出处: <http://zweifymqz.cnblogs.com/>

本文版权归作者和博客园共有, 欢迎转载, 但未经作者同意必须保留此段声明, 且在文章页面明显位置给出原文连接, 否则保留追究法律责任的权利。

分类:  算法—dsu on tree

标签:  dsu on tree

好文要顶

关注我

收藏该文



10

上一篇: [cf600E. Lomsat gelral\(dsu on tree\)](#)

下一篇: [Kruskal重构树入门](#)

posted @ 2018-09-20 20:59 自为风月马前卒 阅读(2391) 评论(4) 编辑 收藏

### 评论列表

#1楼

2019-02-16 14:08 jklover



刚看以为是学业水平考试的题...

支持(2)

反对(0)

#2楼

[楼主] 2019-02-16 20:18 自为风月马前卒



@jklover  
233

支持(0) 反对(0)

#3楼 2019-02-27 21:45 SovietPower



T2可以在这里交2333  
<https://www.luogu.org/problemnew/show/U40581>

支持(1) 反对(0)

#4楼 2019-11-07 22:03 liubainian






STO

支持(0) 反对(0)

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。

Contact with me

-  weibo
-  Github
-  QQ
-  Email