倍增plus差分

From xyw5vplus1

倍增

字面意思，成倍地增长
可以把复杂度的某一维从n降成logn
往往处理区间类的问题，数组的某一维j表示长度为2^j的区间

适合处理多查询少修改的题目，
一般都是预处理，然后高效处理每次查询(在线做法)

RMQ

RMQ(Range Minimum Query)　区间最值查询

ST(Sparse Table 稀疏表)算法

O(nlogn)预处理 O(1)查询

设f[i][j]表示从[i,i+2^j-1]内的最小值

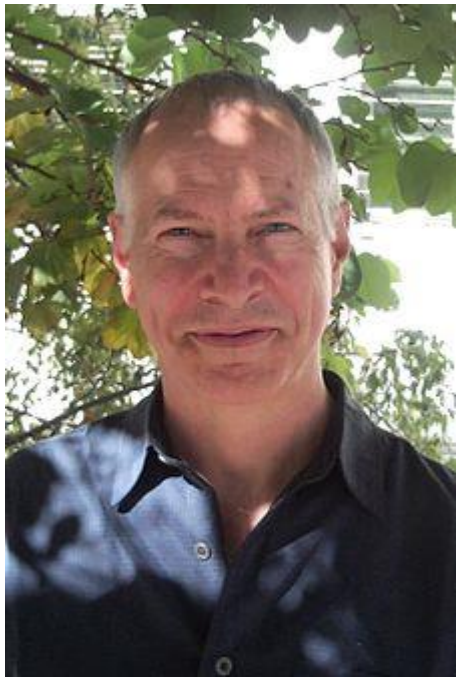
f[i][j]=max(f[i][j-1],f[i+2^(j-1)][j-1]) 画个图理解一下？

Query(L,R)

先查询满足2^k<=(R-L+1)的最大k, k=[log2(R-L+1)]

那么答案就是min(f[L][k],f[R-(2^k)+1][k])

LCA



LCA (Lowest Common Ancestors) 最近公共祖先

离线：Tarjan算法 O(n+q)

在线：倍增做法 O(nlogn) 预处理

O(logn)的时间处理每次查询 已经足够优秀

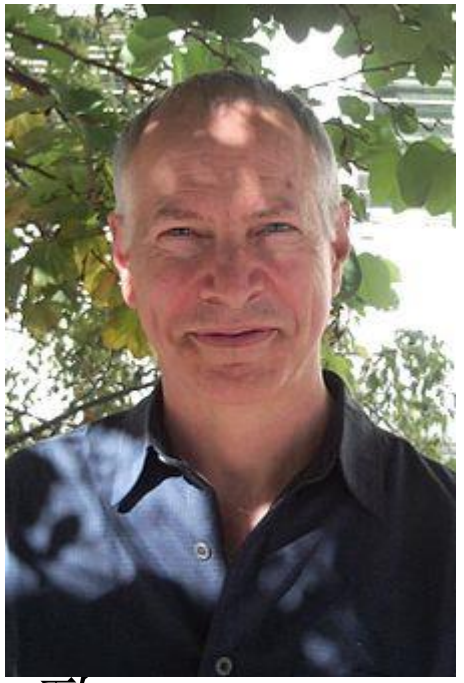
具体做法：维护每个点的第2^i级祖先 father[cur][i]

那么 father[cur][i]=father[father[cur][i-1]][i-1]

用汉字解释的话就是我的第2^i级爸爸是我的2^(i-1)级爸爸的2^(i-1)级爸爸

dfs一遍完成预处理

LCA(倍增做法续)



维护每个点的第2^i级祖先 father[cur][i]

那么 father[cur][i]=father[father[cur][i-1]][i-1]

dfs一遍完成预处理

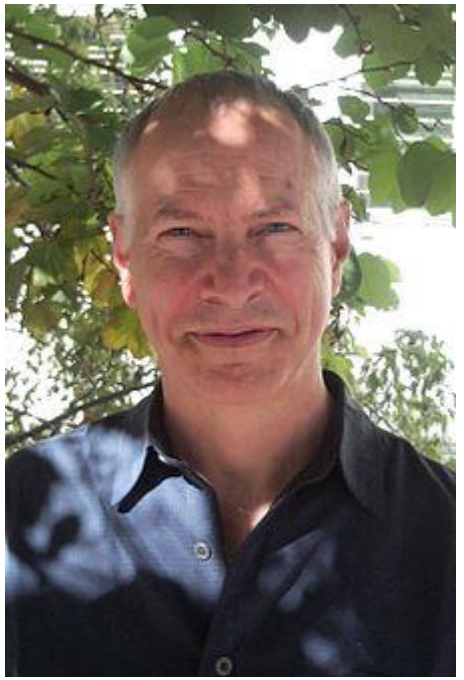
对于每次查询x和y的LCA时，我们先控制x和y的深度一致，然后倍增地往上跳，注意顺序要从大到小，因为不能跳过头

具体来说就是从大到小枚举我当前跳的步长(2^k),

如果father[x][k]!=father[y][k]，那么就往上跳2^k，

否则就不往上跳，因为我们要求的是最近的公共祖先。

LCA(倍增做法续 代码部分)

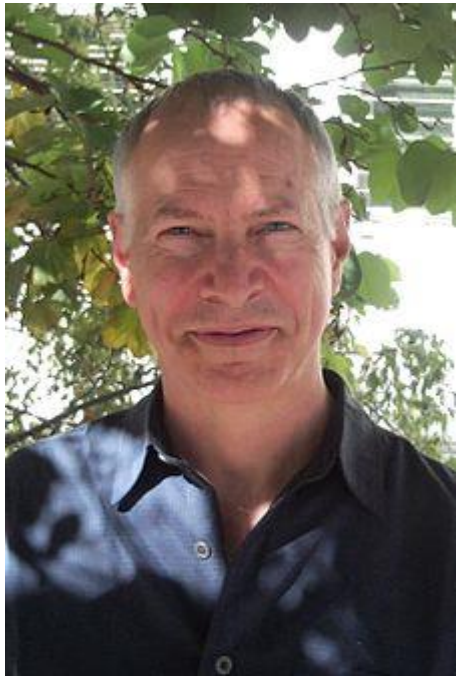


实现的细节:

一个常数优化是预处理一个lg[i]表示log2(i)的整数部分

```
int lca(int x,int y)
        if (dep[x]<dep[y]) swap(x,y);
        while (dep[x]>dep[y]) x=father[x][lg[dep[x]-dep[y]]];
        if (x==y) return x;
        for (int k=lg[dep[x]-dep[y]];k>=0;k--)
                if (father[x][k]!=father[y][k]) {
                        x=father[x][k]; y=father[y][k];
                }
        return father[x][0];
```

LCA与RMQ之间的互相转化



±1 RMQ 要求RMQ的数列中相邻两项的差值只能是1或-1。
利用分块和原先的ST算法可以做到O(n)预处理。

RMQ → LCA 用笛卡儿树(Cartesian Tree) 一种特殊的堆
LCA → ±1 RMQ 对树做一遍dfs，维护一些信息。
上面两种转化的复杂度都是O(n)
这样，对于一般性的RMQ，我们都可以做到O(n)预处理，O(1)查询

LCA(应用篇)

树上两个节点的LCA是这两点之间非常重要的一个连接桥梁。

比如求一棵带权树的树上两点之间距离

很多要处理树上两点之间的路径的一些信息的查询的题目都需要建立在LCA的基础上，而且有些复杂一点的题目需要用类似求LCA的倍增思想维护一些信息。

比如维护树上两点之间路径上的边权最大值

再比如今天的ABCDEFGHIJKL中的若干题不等

希望大家在做题中体会。

差分

能把对原数组一段连续区间的操作转化成差分数组的单点修改

这里的差分指的是广义的差分，不限于相邻两项作差，

还可以相邻两项求xor等等

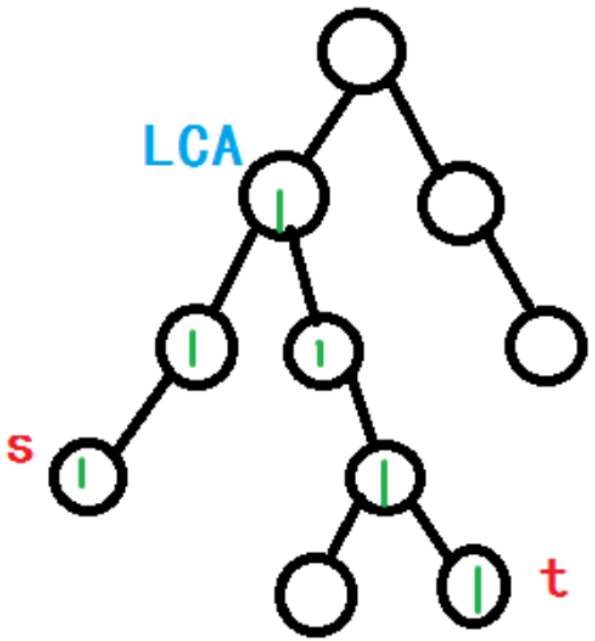
能直接把原先朴素做法的复杂度降低若干个数量级。

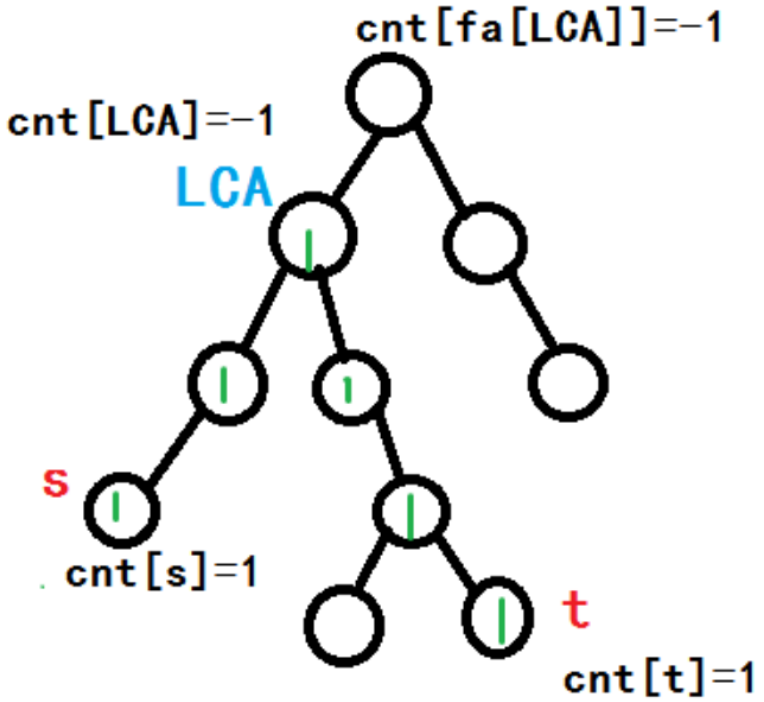
(几维差分就降低几个数量级)

适合处理多修改，少询问(*离线的区间修改*)的问题

树上差分

引入问题：给定若干条树上路径，统计每个点经过了多少次。

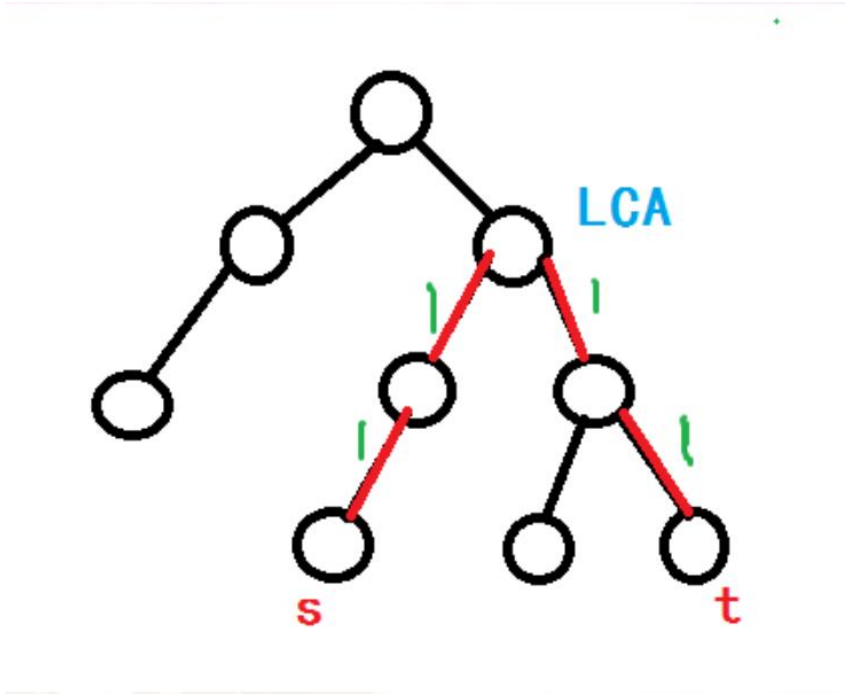


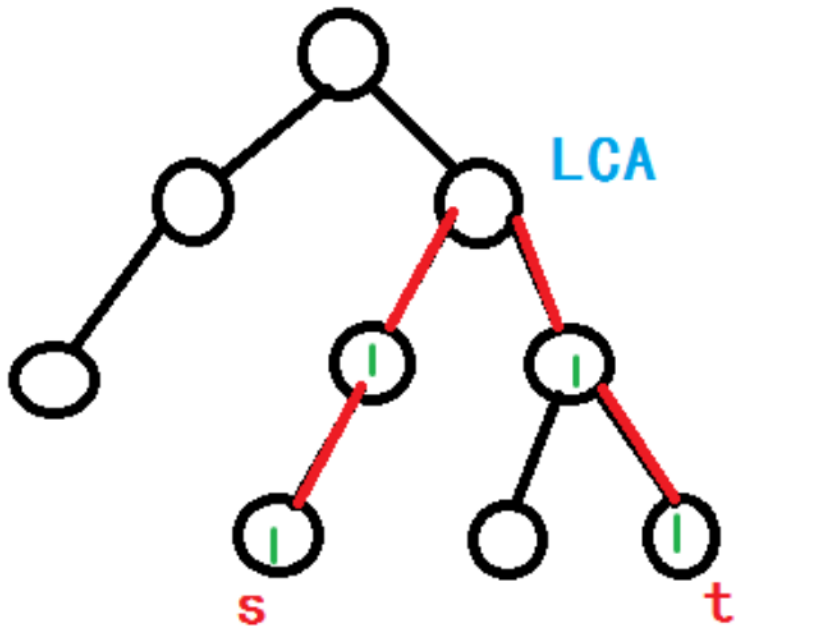


[洛谷博客]

树上差分

引入问题：给定若干条树上路径，统计每条边经过了多少次。





洛谷博客

END……

谢谢大家~~~