Misha and LCP on Tree

题目描述

给定一颗n个节点，且每个节点都有一个字符的树，Q次询问，每次询问求树上两条链构成的字符串的LCP

数据范围

$n \le 3 \times 10^5, Q \le 10^6$

解题过程

我们先考虑，树是一条链的情况

可以发现，我们可以将树写成一个字符串，并且通过后缀数组或者后缀自动机来快速解决每一个询问

那么我们可以通过树链剖分将树分成一些互不相交的链，并且每次询问最多涉及到$\log n$条链

对于所有链的正串与反串合并在一起求一个后缀数组（或建立广义后缀自动机

然后就可以快速的求任意的两个树链的LCP

那么每次将询问的取出，向后跳LCP即可

时间复杂度：$O((n+Q)\log n)$

Organizing a Race

题目大意

给定一个$n+1$个点的道路，并且给定相邻两个点的距离，i与$i+1$的距离为w_i

每个城市有一个加油站，可以提供g_i的油

要求选择两个城市a,b，满足在空油从a出发，能够在中途一直有油并且到底b并返回a

然后你有K升的油，可以放在任意位置，求如何放置能使$b-a+1$最大

数据范围

$n \le 10^5$ ，$w_i, g_i, K \le 10^9$

题解

我们先求出来在不用添加油的情况下，能从i走到哪里，记为$nxt[i]$

然后就可以根据$nxt[i]$建立一棵树，我们称之为$next$树

那么我们遍历这颗$next$树就相当于是枚举以每个位置为开始的情况了

接下来，我们知道，加油一定放在尽可能靠右的位置最好，因为我们还需要往回走

那么也就是放在每个的$nxt[i]-1$的位置上最好

现在，我们求两个数组$pre[i]$与$suf[i]$，其中
$pre[i]=pre[i-1]-w[i]+g[i],suf[i]=suf[i-1]-w[i-1]+g[i]$

可以看出来，$pre[i]$表示从最前向后走到$i+1$剩余的油（可能为负数

可以看出来，$suf[i]$表示从最后向前走到$i-1$剩余的油（可能为负数

那么我们可以知道的是，我们假如需要从$i->nxt[i]$，就需要在$nxt[i]-1$的位置放入
$pre[nxt[i]-1]-pre[i-1]$的油

那么，我知道，从$a->b$所需要的油就是$pre[b-1]-pre[a-1]$

那么我们考虑返程需要的油，也就是需要在最右边的点加入的油满足能开回去

我们假设$suf[i]$在加油的同时已经被修改过了，显然只需要用线段树维护区间加即可

那么返程需要的油也就是：$\max\limits_{k=i}^{j}(suf[k])-suf[j]$

所以我们就只需要要求：$\max\limits_{k=i}^{j}(suf[k])-suf[j]+pre[j-1]-pre[i-1]\le K$即可

那我们设：$p[j]=-suf[j]+pre[j-1]-pre[i-1]$，可以发现，这个$p$不会随着加油减油而变化

那么我们就可以在线段树上维护三个信息：max_suf,max_p以及：$\min\limits_{i=mid+1}^{r}(p[i]+\max\limits_{j=l}^{mid}(suf[j]))$

前两者只需要简单的区间加即可

后者需要在$pushup$的时候递归计算，所以总体时间复杂度为：$O(n\log^2 n)$

Ribbons on Tree

题目描述

给一棵有偶数个点的树，分成$\frac{n}{2}$对，将没对节点之间的最短路径覆盖，求有多少种分配方案满足，树上任意一条边都被覆盖。

数据范围

$n\le 5000$

题解

我们考虑使用容斥解决

对于对于任意一个边集S，考虑设$f(S)$表示不覆盖S的方案数是多少，那么答案也就是
$\sum\limits_{S\subseteq E}(-1)^{|S|}f(S)$。

而对于一个S考虑计算$f(S)$，可以发现，就是将树分成若干个连通块，每个连通块大小为a_i，然后可以知道$f(S)=\prod\limits_{i}[2|a_i](a_i-1)!!$，我们设：$g(i)=[2|i](i-1)!!$

然后的话，考虑求所有的$f(S)$，使用DP解决

设：$f[i][j]$表示i的子树中有j条想要往上走的丝带。

然后的话，考虑正常转移就是一个树上背包问题，复杂度为$O(n^2)$

考虑将两个点匹配的情况，由于我们不需要知道，具体是再哪里两个点匹配上了，假设在i这个位置将不会再有向上走的链了，那么可以知道的是，$f[i][0]=-\sum\limits_{j=1}^{n}f[i][j\times 2]\times g[j\times 2]$

这样表示，我们考虑的这个连通块的已经被分开了，并且算上了该算的贡献

那么，最后的答案就是$-f[1][0]$了