ZJNU 2020-02-04 题解

比赛网址:https://vjudge.net/contest/355151

代码见:banfcc这个账号的提交

A

马拉车裸题

B

单点修改，区间查询的数据结构裸题

C

将所有的串按照字典序排好序后，有某一前缀的是一段连续的区间，求出所有串的字典序，按照字典序大小建线段树操作3就是区间查询$(length, rank)$长度和字典序排名的二元组的最小值,操作1，2都是单点修改

刘浩的提供题解

题意

三种操作，一共有Q个：

- $1\ X$：在字典里插入字符串 X
- $2\ X$：在字典里删除字符串 X
- $3\ X$：输出字典中最短的且前缀是 X 的下标。如果有多个字符串，输出字典序最小的。

字符串的下标是指该串被插入时的时间。

输入中所有的字符串长度和不超过10^6

题解

第一眼感觉是可持久化AC自动机？不会啊。然后就看到了这个条件：所有的字符串长度和不超过 10^6 ，决定乱搞。

对每个长度开一个$map < hash, set > [i]$，表示存在一个 set 的字符串满足长度为 i 的前缀哈希值是 $hash$。

set 里记录一下字符串的信息，比如长度啊，字典序啊，下标啊。这个字典序我的求法就比较夸张，直接全部读进来，离散排个序。

然后三种操作只要对应的把 map 和 set 维护好就行了。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=5e5+7;
int q;
int op[N];
string s[N],ss[N];
```

```
int px[N];
int er[N];
struct Node{
    int id,len,p;
    bool operator<(const Node k)const{
        if(len==k.len) return p<k.p;
        return len<k.len;
    }
};
unordered_map<unsigned long long,set<Node>>mp[N*10];
int main(){
    scanf("%d",&q);
    for(int i=1;i<=q;i++){
        scanf("%d",&op[i]);
        if(op[i]==2) scanf("%d",&er[i]),ss[i]=ss[er[i]];
        else cin>>ss[i];
        s[i]=ss[i];
    }
    sort(ss+1,ss+1+q);
    for(int i=1;i<=q;i++){
        px[i]=lower_bound(ss+1,ss+1+q,s[i])-ss;
    }
    for(int i=1;i<=q;i++){
        if(op[i]==1){
            unsigned long long h=0;
            for(int j=0;j<s[i].length();j++){
                h=h*233+s[i][j]-'a';
                mp[j][h].insert((Node){i,s[i].length(),px[i]});
            }
        }
        else if(op[i]==2){
            unsigned long long h=0;
            h=0;
            for(int j=0;j<s[i].length();j++){
                h=h*233+s[i][j]-'a';
                mp[j][h].erase((Node){er[i],s[i].length(),px[i]});
            }
        }
        else{
            unsigned long long h=0;
            for(int j=0;j<s[i].length();j++){
                h=h*233+s[i][j]-'a';
            }
            if(mp[s[i].length()-1][h].empty()) printf("-1\n");
            else printf("%d\n",(*mp[s[i].length()-1][h].begin()).id);
        }
    }
}
```

D

给n个圆，一个初始点，求在初始点的圆最小的半径能够吞掉n个圆

吞掉的规则是如果和初始点的圆有公共部分，相切也算，就会被吞掉，圆半径增加吞掉的圆的半径

设dis_i为第i个圆的圆心和初始点的距离，r_i为半径，显然按$dis_i - r_i$排序贪心即可

E

$nlogn$预处理每个数的因子和，再预处理答案即可

F

扫描线或者离散化后用什么数据结构都可以

G

简单模拟

H

dijkstra算法的一点点变形

I

一开始全部方格都无效，按照权值大小从大到小依次加入，用并查集处理有效部分的联通块大小，维护所有联通块大小的最大值，当最大值大于S时就找到了答案

或者二分再用类似的做法维护联通块大小

J

树形dp

$dp[u][c]$表示以u这个点连向父亲的边的颜色是 c的的方案数

$f[u][i]$表示u这个点的儿子们连向父亲的那条边占用的颜色情况壮压后为i的方案数

儿子的dp, f之间用类似背包的转移，这部分的复杂度是$O(n2^55)$

f把所有的儿子的情况考虑完后，就可以根据f占用的颜色和连向父亲的边的颜色得到自己dp，这部分也是$O(n2^55)$

时间复杂度是$O(n2^55)$，空间复杂度是$O(n2^45)$

当然时间复杂度可以做到$O(n2^45)$只枚举那些有效的部分，也不是很有必要

K

简单dp

$dp[i][j] = min(dp[i-1][j-1], dp[i][j-1], dp[i+1][j-1]) + a[i][j]$

再处理下边界情况即可