

Codeforces Round #706 (Div. 1)

[比赛链接](#)

B. Tree Array

题意

给定 n 个点的树，点编号分别为 $1 \sim n$

初始时等概率随机选择一个点，接下来每次操作从未选择的且与已经选择的点相连的点中等概率随机选择一个点。

根据选择的顺序可以得到由点的编号构成的序列，问序列逆序对的期望个数。

题解

首先枚举初始选择点，并将初始选择点作为根节点。

不难发现，每个节点只有在所有祖先节点被选取后才有机会选取。

同时对任意点对 (u,v) 当 $p = \text{LCA}(u,v)$ 被选取后，他们的其余祖先节点被选取的概率总是相等的。

于是不妨设 $u > v, d_1 = d_u - d_p, d_2 = d_v - d_p$ 于是 (u,v) 构成逆序对的概率等价于如下模型：

两个袋子中分别有 d_1, d_2 个球，每次有 t 概率从某个袋子中选一个球，也有 $1-2t$ 概率无事发生，问 d_2 个球的袋子先被取空的概率。

不难发现 $\text{dp}(d_1, d_2) = \frac{\text{dp}(d_1-1, d_2) + \text{dp}(d_1, d_2-1)}{2}$ 然后暴力统计固定根每个点对贡献即可，时间复杂度 $O(n^3)$

```
const int MAXN=205,mod=1e9+7;
int dp[MAXN][MAXN];
int quick_pow(int a,int k){
    int ans=1;
    while(k){
        if(k&1)ans=1LL*ans*a%mod;
        a=1LL*a*a%mod;
        k>>=1;
    }
    return ans;
}
struct Edge{
    int to,next;
}edge[MAXN<<1];
int head[MAXN],edge_cnt;
```

```
void Insert(int u,int v){
    edge[++edge_cnt]=Edge{v,head[u]};
    head[u]=edge_cnt;
}
vector<int> ch[MAXN];
int dep[MAXN],ans;
void dfs(int u,int fa){
    dep[u]=dep[fa]+1;
    ch[u].clear();
    ch[u].push_back(u);
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        if(v==fa)continue;
        dfs(v,u);
        _for(j,0,ch[u].size())
        _for(k,0,ch[v].size()){
            int x=ch[u][j],y=ch[v][k];
            int d1=dep[max(x,y)]-dep[u],d2=dep[min(x,y)]-dep[u];
            ans=(ans+dp[d1][d2])%mod;
        }
        _for(j,0,ch[v].size())
        ch[u].push_back(ch[v][j]);
    }
}
int main()
{
    int inv2=quick_pow(2,mod-2);
    _for(i,1,MAXN)dp[0][i]=1;
    _for(i,1,MAXN)_for(j,1,MAXN)
    dp[i][j]=1LL*(dp[i-1][j]+dp[i][j-1])*inv2%mod;
    int n=read_int();
    _for(i,1,n){
        int u=read_int(),v=read_int();
        Insert(u,v);
        Insert(v,u);
    }
    _rep(i,1,n)
    dfs(i,0);
    enter(1LL*ans*quick_pow(n,mod-2)%mod);
    return 0;
}
```

C. Converging Array

题意

一开始有 $a[1 \sim n], b[1 \sim n-1]$ 接下来有无限次操作。

每次操作随机选择一个 $1 \leq i \leq n-1$ $a_i = \min\left(a_i, \frac{a_i + a_{i+1} - b_i}{2}\right)$, $a_{i+1} = \max\left(a_i, \frac{a_i + a_{i+1} + b_i}{2}\right)$

可以证明无限次操作后序列 $a[1 \sim n]$ 收敛于固定值。

每次询问给定 x 问有多少序列 $a[1 \sim n]$ 满足：

1. $0 \leq a_i \leq c_i$
2. 最终收敛后 $a_1 \geq x$

题解

设最终得到序列为 $f[1 \sim n]$ 观察发现，每次操作不改变 $\sum a_i$ 且最后有 $f_{i+1} - f_i \geq b_i$

设 $s a_i = \sum_{j=1}^i a_j$, $s b_1 = 0$, $s b_{i+1} = \sum_{k=1}^j b_k$, $s s b_i = \sum_{j=1}^i c_j$
 $s b_i$, $s c_i = \sum_{j=1}^i c_j$

不难发现最后形式为 $f_1 + s b_1, f_1 + s b_2, f_1 + s b_3 \dots f_1 + s b_k, f_{k+1} \dots$ 其中 $f_t \geq f_1 + s b_t$ ($t > k$)

同时有 $\sum_{i=1}^k f_i + s b_i = \sum_{i=1}^k a_i$ 且 $\sum_{i=1}^t f_i + s b_i = \sum_{i=1}^t a_i \geq \sum_{i=1}^t a_i$ ($t \neq k$)

于是有 $f_1 = \min_{i=1}^n \left(\frac{s a_i - s s b_i}{i} \right)$ 枚举询问的 x 对每个 x 保证 $\frac{s a_i - s s b_i}{i} \geq x$ 即可计算答案个数。

设 $dp(i, s)$ 表示 $\sum_{j=1}^i a_j = s$ 的情况 $m = \max c_i$ 利用差分可以 $O(n^2 m)$ 计算答案。

最后有 $\min\left(\frac{-s s b_i}{i}\right) \leq f_1 \leq \min\left(\frac{s c_i - s s b_i}{i}\right)$ 于是有 $\max(f_1) - \min f_1 \leq m$

即有效的 x 只有 $O(m)$ 个，最终时间复杂度 $O\left(n^2 m^2\right)$

```
const int MAXN=105,MAXV=1e4+5,mod=1e9+7,inf=1e9;
int b[MAXN],c[MAXN],dp[MAXN][MAXV],ans[MAXN],n;
int solve(int x){
    mem(dp,0);
    dp[0][0]=1;
    _rep(i,1,n){
        _for(j,1,MAXV)
            dp[i-1][j]=(dp[i-1][j]+dp[i-1][j-1])%mod;
        _for(j,max(x*i+b[i],0),MAXV){
            if(j-c[i]>0)
                dp[i][j]=(dp[i-1][j]-dp[i-1][j-c[i]-1]+mod)%mod;
            else
                dp[i][j]=dp[i-1][j];
        }
    }
    int ans=0;
    _for(i,0,MAXV)
```

```
ans=(ans+dp[n][i])%mod;
return ans;
}
int Div(int a,int b){
if(a<0)
return (a-b+1)/b;
else
return a/b;
}
int main()
{
n=read_int();
_rep(i,1,n)c[i]=read_int();
_rep(i,2,n)b[i]=read_int()+b[i-1];
_rep(i,2,n)b[i]=b[i]+b[i-1];
int lef=inf,rig=inf,s=0;
_rep(i,1,n){
s+=c[i];
lef=min(lef,Div(-b[i],i));
rig=min(rig,Div(s-b[i],i));
}
_rep(i,lef,rig)
ans[i-lef]=solve(i);
int q=read_int();
while(q--){
int x=read_int();
x=min(max(lef,x),rig+1);
enter(ans[x-lef]);
}
return 0;
}
```

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_728_div_1&rev=1625538563 

Last update: 2021/07/06 10:29