pk赛题解

psc233的舞会

一个一维随机游走题，记$f[n]$为当前有n个女孩,n+2个男孩，舞会永不结束的概率 则 $f[n]=\frac{n}{2n+2}f[n-1]+\frac{n+2}{2n+2}f[n+1]\quad f[0]=0,f[+\infty]=1$

化简上述式子可得$n(f[n]-f[n-1])=(n+2)(f[n+1]-f[n])$

$f[n]-f[n-1]=\frac{2}{n(n+1)}f[1]$

裂项可推得$f[1]=\frac{1}{2}$，所以$f[n]=\frac{n}{n+1}$

取石子游戏

朴素的想法就是$f[i][j][k]$,表示选到第i堆石子,剩下的石子个数异或值是j,选择堆数对d取模后的结果是k的方案数.

首先我们变换选择石子顺序并不会影响最后的答案,我们对$a[1..n]$从大到小排序,可以发现如果 $a[i]<2^L$,那么在之后的转移中第L位就不会发生变化了

于是我们可以把$a[i]$分成log段进行处理,控制转移方程枚举的上限

复杂度为$O(d*\sum a[i])$

```
#include<cstdio>
#include<algorithm>
using namespace std;
const int maxn=1100005,tt=1e9+7;
int f[2][10][maxn],n,D,a[maxn];
inline int _read(){
    int num=0;char ch=getchar();
    while(ch<'0'||ch>'9') ch=getchar();
    while(ch>='0'&&ch<='9') num=num*10+ch-48,ch=getchar();
    return num;
}
bool cmp(int x,int y){return x>y;}
int main(){
    n=_read();D=_read();
    for (int i=1;i<=n;i++) a[i]=_read();
    sort(a+1,a+1+n,cmp);f[0][0][0]=1;
    for (int i=1;i<=n;i++){
        int L=20;while((1<<L)>a[i]) L--;L++;
        for (int j=0;j<D;j++)
        for (int k=0;k<(1<<L);k++) f[i&1][j][k]=0;
        for (int j=0;j<D;j++)
        for (int k=0;k<(1<<L);k++) if (f[1-(i&1)][j][k]){
            (f[i&1][j][k^a[i]]+=f[1-(i&1)][j][k])%=tt;
            (f[i&1][(j+1)%D][k]+=f[1-(i&1)][j][k])%=tt;
        }
```

```
    }
    if (n%D==0) f[n&1][0][0]=(f[n&1][0][0]-1+tt)%tt;
    printf("%d\n",f[n&1][0][0]);
    return 0;
}
```

树

做法很多,比较简单的写法是分治区间$[L,R]$,对于中点m,我们考虑经过m的区间,我们以m为根做DFS,对于每个点统计到根的路径上点的编号的最大值和最小值,令Q_i表示区间$[i,m]or[i,m]$的区间的点到根的路径上权值最大值,P_i表示最小值.显然一个区间$[i,j]$可行的必要条件是$P_i >= i, Q_j <= j$,可以发现随着i的减小,满足条件的最小的j和最大的j都是单调的,所以可以用$Two-pointers$解决,复杂度$O(nlogn)$

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;
const int maxn=300005,INF=1073741823;
inline int _read(){
    int num=0;char ch=getchar();
    while(ch<'0'||ch>'9') ch=getchar();
    while(ch>='0'&&ch<='9') num=num*10+ch-48,ch=getchar();
    return num;
}
int lnk[maxn],tot,son[2*maxn],nxt[2*maxn];
int T,n,f[maxn],g[maxn],a[maxn];
long long ans;
bool vis[maxn];
void add(int x,int y){nxt[++tot]=lnk[x];lnk[x]=tot;son[tot]=y;}
void DFS(int x,int L,int R){
    vis[x]=1;
    for (int j=lnk[x];j;j=nxt[j])
    if (!vis[son[j]]&&son[j]>=L&&son[j]<=R){
        f[son[j]]=max(f[son[j]],f[x]);g[son[j]]=min(g[son[j]],g[x]);
        DFS(son[j],L,R);
    }
}
void solve(int L,int R){
    if (L==R){ans++;return;}
    int mid=L+(R-L>>1);
    for (int i=L;i<=R;i++) f[i]=g[i]=i;
    DFS(mid,L,R);
    for (int i=L;i<=R;i++){
        if (!vis[i]) f[i]=INF,g[i]=-INF;
        vis[i]=0;
    }
    for (int i=mid+1;i<=R;i++) f[i]=max(f[i-1],f[i]),g[i]=min(g[i-
1],g[i]),a[i]=f[i]<=i;
    for (int i=mid-1;i>=L;i--) f[i]=max(f[i+1],f[i]),g[i]=min(g[i+1],g[i]);
    int now=0,i2=mid+1,i1=mid;
    for (int i=mid;i>=L;i--) if (g[i]>=i){
```

```
            while(i1<R&&g[i1+1]>=i) now+=a[++i1];
            while(i2<=i1&&f[i]>i2) now-=a[i2++];
            ans+=now;
        }
        solve(L,mid);solve(mid+1,R);
    }
    void work(){
        memset(lnk,0,sizeof(lnk));tot=0;
        n=_read();
        for (int i=1;i<n;i++){
            int x=_read(),y=_read();
            add(x,y);add(y,x);
        }
        ans=0;solve(1,n);
        printf("%lld\n",ans);
    }
    int main(){
        T=_read();while(T--) work();
        return 0;
    }
```

psc233的序列

%10: 2^n直接爆搜即可

%30： f[l][r][k]表示序列起点为l，终点为r，选了k个数的答案 $f[l][r][k]=min(max(f[l][x][k-1],a_x+a_r)),l<=x<r$ 最后还要与a_l+a_r取max

%100: 做法一：考虑二分答案x,贪心全选值$<=\lfloor\frac{x}{2}\rfloor$的数 对于已选的两个数中间最多选一个$>\lfloor\frac{x}{2}\rfloor$的数,注意首尾 该做法时间复杂度为$O(nlogn)$

做法二：考虑和最大的一对相邻的数，则把它们中的较大值删掉一定最优。用堆维护即可。时间复杂度$O(nlogn)$。

做法三：考虑二分答案x，显然最小值必然选，将该序列平移使得最小值在第一位。考虑dp[i]表示选的序列终点为i,起点为1，满足条件的能够最多选的数的个数。

$dp[i]=max(dp[j]+1),a_j+a_i<=x,j<i\quad dp[1]=1$ 该dp可以用数据结构优化，时间复杂度为$O(nlog^2n)$ 注意常数

注意如果你二分写的是$(l+r)/2$那么要开long long,因为答案的值域为2e9

upd:对于这道题数据水导致一些乱搞做法过的问题我深表歉意

菜地

最大子矩阵裸题

传送门

将AB两种形态拆点后建边，跑最短路