

博弈题杂

博弈题杂

2022牛客多校2 C - Link with Nim Game

2022牛客多校7 K-Great Party (莫队+Nim)

2022牛客多校2 C - Link with Nim Game

给定 n 堆石子，每个回合，玩家选择一堆石子，移除任意正整数个。如果玩家处于必胜态，则会用最优策略尽快结束游戏，否则会用最优策略推迟游戏的结束。问最后游戏能进行多少回合以及Alice最开始应该拿多少石子。

一般Nim游戏结论：异或和为0必败，否则必胜。那么先手把必败态转移为必胜态，只需要将异或和为1的数位变成另即可。①考虑必败的情况下，我们要尽量延长回合数，实际上任取一个即可，因为对手必须把最低位异或和变为0，因此也只能减一个。特殊地，初始操作时如果有某一堆存在两个连续的1会使得使得对手可以取多个石子，用一个函数check一下即可。考虑必胜的情况下，对方会让我们进入只能取一个的局面，所以就是问我们第一步最多能取多少个，有多少种取法，因此记录最高的lowbit($a[i]$)使得 $s^{\text{lowbit}(a[i])} < a[i]$ 即可。

```
#include<bits/stdc++.h>
#define lb(x) (x&(-x))
#define int long long
using namespace std;
const int N=1e5+7;

bool cmp(int x,int y){return lb(x)<lb(y);}

int n,a[N],pre[N],suf[N],T,s,ans,sum,mx;

signed main(){
    ios::sync_with_stdio(0);
    cin.tie(0);cout.tie(0);
    cin>>T;
    while(T--){
        cin>>n;
        sum=0;s=0;
        for(int i=1;i<=n;++i){
            cin>>a[i];
            sum+=a[i];
            s^=a[i];
        }
        if(!s){
            ans=0;
            sort(a+1,a+1+n,cmp);
            pre[0]=0;suf[n+1]=0;
            for(int i=1;i<=n;++i) pre[i]=pre[i-1]|a[i];
            for(int i=n;i;--i) suf[i]=suf[i+1]|a[i];
            for(int i=1;i<=n;++i){
                int j=i;
                while(j<n&&lb(a[j+1])==lb(a[i])) ++j;
            }
        }
    }
}
```

```

        if(!((pre[i-1]/lb(a[i]))&1)&&!(((suf[j+1]/lb(a[i]))&1))) {
            ans+=j-i+1;
        }
        i=j;
    }
    printf("%lld %lld\n",sum,ans);
}
else{
    mx=0,ans=1;
    for(int i=1;i<=n;++i){
        if((a[i]^s)>a[i]) continue;
        int x=a[i]-(s^a[i]);
        if(x>mx) mx=x,ans=1;
        else if(x==mx) ++ans;
    }
    printf("%lld %lld\n",sum-mx+1,ans);
}
}
return 0;
}

```

2022牛客多校7 K-Great Party (莫队+Nim)

给定 n 堆石子，每次操作有两步：①选一堆非空石子堆，移去正整数个石子；②保持石堆中剩下的石头不动或者合并到另一堆中。

```

#include<bits/stdc++.h>
#define int long long
using namespace std;
const int N=1e6+7;

struct Q{
    int l,r,len,id,be1;
}s[N];

int n,m,a[N],ans[N],now=0,cnt[2][N<<2];

bool cmp(Q A,Q B){
    if(A.be1^B.be1) return A.be1<B.be1;
    return (A.be1&1)?A.r>B.r:A.r<B.r; //奇偶性优化
}

void add(int x){
    int p=(x&1),y=a[x];
    now+=cnt[p][y];
    cnt[p][y]++;
}

void del(int x){
    int p=(x&1),y=a[x];
    --cnt[p][y];
    now-=cnt[p][y];
}

signed main(){
    ios::sync_with_stdio(0);

```

```

cin.tie(0);cout.tie(0);
cin>>n>>m;
int bk=sqrt(n);
a[0]=0;
for(int i=1;i<=n;++i){
    cin>>a[i];
    a[i]=(a[i]-1)^a[i-1];
}
for(int i=1;i<=m;++i){
    cin>>s[i].l>>s[i].r;
    s[i].l--;
    s[i].id=i;s[i].len=s[i].r-s[i].l+1;
    s[i].bel=(s[i].l-1)/bk+1;
}
stable_sort(s+1,s+1+m,cmp);
int L=1,R=0;
for(int i=1;i<=m;++i){
    while(L>s[i].l) add(--L);
    while(R<s[i].r) add(++R);
    while(R>s[i].r) del(R--);
    while(L<s[i].l) del(L++);
    int all=(s[i].len*(s[i].len-1))/2;
    ans[s[i].id]=all-now;
}
for(int i=1;i<=m;++i) cout<<ans[i]<<"\n";
return 0;
}

```