2020/05/02-2020/05/08周报

团队训练

本周无团队训练

李元恺

专题

没有专题

比赛

没有比赛

题目

TJOI2019 唱、跳□rap□篮球

分类：生成函数□FFT

一句话题意：有四类人排队，每类人分别喜欢唱、跳□rap□篮球，分别有a,b,c,d个人，队伍长度n□如果任意k□k□k+1,k+2,k+3四个位置上的人依次喜欢唱、跳□rap□篮球，则不合法，求合法的排列方法数 mod 998244353□n□a,b,c,d⇐1e3

解法：注意任意两个四人组不可能有交，分别求至少包含1，2，…，个四人组不合法，求法使用指数型生成函数，最后容斥

```
#include <bits/stdc++.h>
using namespace std;
const int N = 4040;
long long a[N],b[N],nn = 1,rev[N],w1[N],w2[N];
const int mod = 998244353;

inline int power(int di,int ci) {
    int ret = 1;
    while (ci) {
        if (ci&1)
            ret = (long long)ret*di%mod;
        di = (long long)di*di%mod;
        ci >>= 1;
```

```
    }
    return ret;
}
inline long long inv(int x) {
    return power(x,mod-2);
}
inline void NTT(long long *x,int I) {
    int i,j;
    long long t0,t1,*w;
    int k;
    for (i = 0;i < nn; i++)
        if (rev[i] > i)
            swap(x[rev[i]],x[i]);
    w = (I == 1?w1:w2);
    for (i = 1;i < nn; i <<= 1) {
        for (j = 0;j < nn; j += (i<<1)) {
            for (k = 0;k < i; k++) {
                t0 = x[j|k],t1 = (long long)w[i|k]*x[i|j|k]%mod;
                x[j|k] = (t0+t1)%mod;
                x[i|j|k] = ((t0-t1)%mod+mod)%mod;
            }
        }
    }
    if (I == -1)
        for (int i = 0;i < nn; i++)
            x[i] = (long long)x[i]*inv(nn)%mod;
}
int half;
int aa,bb,cc,dd,n;
void calc() {
    for (int i = 0;i < half; i++)
        w1[i|half] = power(3,(mod-1)/nn*i);
    for (int i = half-1;i>0; --i)
        w1[i] = w1[i<<1];
    for (int i = 1;i < nn; i++)
        w2[i] = inv(w1[i]);
    NTT(a,1);
    NTT(b,1);
    for (int i = 0;i < nn; i++)
        a[i] = (long long)b[i]*a[i]%mod;
    NTT(a,-1);
    for (int i = n+1;i <= nn; i++)
        a[i] = 0;
}
long long njc[1010];
inline void work(int p) {
    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    for (int i = 0;i <= min(aa-p,n); i++)
        a[i] = njc[i];
```

```
    for (int i = 0;i <= min(bb-p,n); i++)
        b[i] = njc[i];
    calc();
    memset(b,0,sizeof(b));
    for (int i = 0;i <= min(cc-p,n); i++)
        b[i] = njc[i];
    calc();
    memset(b,0,sizeof(b));
    for (int i = 0;i <= min(dd-p,n); i++)
        b[i] = njc[i];
    calc();
}

long long C[1010][1010];
long long f[1010];
int main() {
    scanf("%d%d%d%d%d",&n,&aa,&bb,&cc,&dd);
    C[0][0] = 1;
    for (int i = 0;i <= 1000; i++) {
        C[i][i] = C[i][0] = 1;
        for (int j = 1;j < i; j++)
            C[i][j] = (C[i-1][j]+C[i-1][j-1])%mod;
    }
    njc[0] = 1;
    while (nn <= n+n)
        nn <<= 1;
    half = nn/2;
    for (int i = 1;i < nn; i++)
        rev[i] = (rev[i>>1]>>1)|((i&1)?half:0);
    for (int i = 1;i <= n; i++) {
        njc[i] = njc[i-1]*inv(i)%mod;
    }
    long long ans = 0;
    for (int i = 0;i <= n/4; i++) {
        if (i > aa || i > bb || i > cc || i > dd)
            break;
        work(i);
        f[i] = a[n-4*i]*inv(njc[n-4*i])%mod*C[n-3*i][i]%mod;
        if (i&1)
            ans -= f[i];
        else
            ans += f[i];
    //    cout<< i << " " << f[i] << endl;
    }
    for (int i = n/4;~i; i--) {
        for (int j = i+1;j <= n/4; j++)
            (f[i] -= f[j]*C[j][i]) %= mod;
    }
    f[0] += mod;
    f[0] %= mod;
    ans %= mod;
```

```
    ans += mod;
    ans %= mod;
//    cout << ans << endl;
    printf("%lld",f[0]);
    return 0;
}
```

姜维翰

专题

没有专题

比赛

没有比赛

题目

cf goodbye 2018 E

题意：一个n点无向图，给出n-1点的度数，问第n个点的所有可能度数，无解输出-1

n=500000

解法：我们可以很容易知道答案的奇偶性，此外若a和b都可行a<b则对于a<c<b且c和ab奇偶性相同，那么c一定可行

所以只要找出上下界就行

原题中给出了一个定理，利用这个定理以及握手定理可以判定给定的度数是否合法

看上去不能二分，但是这个定理在判出非法的同时你还可以知道是偏大还是偏小，所以可以先二分找一个可靠解，再对上下界二分

```
#include<bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef double db;
typedef complex<double> cp;
typedef pair<int,int> pll;

const int maxn=(int)5e5+9;
```

```
const int maxm=(int)1e6+9;
const ll mod=(ll)998244353;
const db pi=acos(-1);
const db eps=1e-15;

#define dbg(x) cerr<<#x<<" is "<<x<<endl;

ll e[maxn];
ll tmp[maxn];
ll sur[maxn];
ll n;

int ck(ll v){
    int p=0;
    int fl=0;
    int pos;
    for(int i=0;i<=n;i++){
        if(!fl&&(p==n||v<=e[0]||(v>e[p-1]&&v<=e[p]))){
            fl=1;
            tmp[i]=v;
            pos=i;
        }else{
            tmp[i]=e[p];
            p++;
        }
    }
    sur[n+1]=0;
    for(int i=n;i>=0;i--){
        sur[i]=sur[i+1]+tmp[i];
        //printf("$%d\n",sur[i]);
    }
    for(int i=n-1;i>=0;i--){
        int pp=upper_bound(tmp,tmp+n+1,n-i)-tmp;
        pp=min(pp,i);
        //printf("# %d %lld %d\n",i,sur[i+1],pp);
        if(sur[i+1]>(n-i)*(n-i-1)+sur[0]-sur[pp]+(n-i)*(i-pp+1)){
            if(i<=pos)return 1;
            else return -1;
        }
    }
    return 0;
}

void init(){
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%lld",&e[i]);
    }
    sort(e,e+n);
}
```

```
int main(){
    init();
    ll sum=0;
    for(int i=0;i<n;i++){
        sum+=e[i];
    }
    ll bg,ed;
    bg=0;
    ed=((ll)n)*(n+1)-sum;
    ed=min(ed,n);
    ll mid;
    while(bg<ed){
        mid=(bg+ed)/2;
        int f=ck(mid);
        //dbg(f);
        //dbg(mid);
        if(f==-1){
            bg=mid+1;
        }else if(f==1){
            ed=mid-1;
        }else{
            break;
        }
    }
    if(bg==ed&&ck(bg)!=0){
        printf("-1\n");
        return 0;
    }
    //dbg(mid);
    ll m1=mid;
    while(bg<m1){
        ll mm=(bg+m1)/2;
        int f=ck(mm);
        if(f){
            bg=mm+1;
        }else{
            m1=mm;
        }
    }
    ll m2=mid;
    while(m2<ed){
        ll mm=(m2+ed+1)/2;
        int f=ck(mm);
        if(f){
            ed=mm-1;
        }else{
            m2=mm;
        }
    }
    ll eo=sum&1;
```

```
    for(ll i=bg;i<=ed;i++){
        if(eo==(i&1)){
            printf("%lld ",i);
        }
    }
    cout<<endl;
```

袁熙

专题

没有专题

比赛

没有比赛

题目

CF1344C Quantifier Question DFS

理解一下题意：给E,V~2e5的图，若无环，求有多少点，满足是其所在连通块上点编号最小的点

花的时间有点久，写之前想的不太充分，只考虑了后面的点 实际解法：做正向和逆向的拓扑排序，确定点是否为编号最小

```
#include <bits/stdc++.h>
#define ll long long
#define tmp(x) std::cout<<"& "<<(x)<<" &\n"
#define rep(i,a,b) for(int i=(a);i<=(b);++i)
#define per(i,a,b) for(int i=(a);i>=(b);--i)
using namespace std;

const int maxn=2e5+100;
const int mo=998244353;

int ck[maxn],deg[maxn],vis[maxn],vs[maxn],vss[maxn],rdeg[maxn];
vector<int> g1[maxn],g2[maxn];
int n,m,u,v,pt;
inline int read(){
    int x=0,f=1;
    char c=getchar();
    while(c>'9'||c<'0'){
        if(c=='-')f=-1;
        c=getchar();
```

```
    }
    while(c>='0'&&c<='9'){
        x=x*10+c-'0';c=getchar();
    }
    return x*f;
}


queue<int> qq;
queue<int> q;
void topo(){
    for(int i=1;i<=n;++i){
        vs[i]=vss[i]=i;
        if(!deg[i])vis[i]=1,q.push(i);
        if(!rdeg[i])qq.push(i);
    }
    while(!q.empty()){
        int x=q.front();q.pop();
        for(int i=0;i<g1[x].size();++i){
            vs[g1[x][i]]=min(vs[x],vs[g1[x][i]]);
            if(--deg[g1[x][i]]==0)vis[g1[x][i]]=1,q.push(g1[x][i]);
        }
    }
    while(!qq.empty()){
        int x=qq.front();qq.pop();
        for(int i=0;i<g2[x].size();++i){
            vss[g2[x][i]]=min(vss[x],vss[g2[x][i]]);
            if(--rdeg[g2[x][i]]==0)qq.push(g2[x][i]);
        }
    }
    rep(i,1,n)if(!vis[i]){
            printf("-1\n");return;
        }
    rep(i,1,n)if(vs[i]==i&&vss[i]==vs[i])++pt,ck[i]=1;

    printf("%d\n",pt);
    rep(i,1,n){
        if(ck[i])printf("A");
        else printf("E");
    }
    printf("\n");
}
int main() {
    freopen("in.txt","r",stdin);
    n=read(),m=read();
    rep(i,1,m){
        u=read(),v=read();
        g1[u].push_back(v),g2[v].push_back(u);
        ++deg[v],++rdeg[u];
    }
```

```
    topo();
    return 0;
}
```

本周推荐

李元恺

推荐后缀数组

袁熙

姜维翰