

# template

## 写在前面

### 基础模版

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
#define OPFI(x) freopen(#x".in", "r", stdin);\
                freopen(#x".out", "w", stdout)
#define REP(i, a, b) for(int i=(a); i≤(b); ++i)
#define REPd(i, a, b) for(int i=(a); i≥(b); --i)
inline ll rd(){
    ll r=0, k=1; char c;
    while(!isdigit(c=getchar())) if(c=='-') k=-k;
    while(isdigit(c)) r=r*10+c-'0', c=getchar();
    return r*k;
}
int main(){
    return 0;
}
```

### vimrc

```
syntax on
set ts=4
set expandtab
set autoindent
set cindent
set shiftwidth=4
set nu
set softtabstop=4
set smartindent
set showmatch
set ruler
set mouse=a
inoremap <F1> <esc>:w<CR>
inoremap <F5> <esc>:below term<CR>
nmap <F1> :w<CR>
nmap <F5> :below term<CR>
colo habamax
set title
set shell=powershell
```

```
set wim=list
set backspace=indent,eol,start
set nocompatible
```

## 数据结构

### zkw 线段树

单点修 区间查

```
ll s[N<<2], a[N];
int M;

ll f(ll x, ll y){
    return x+y; // 改这
}

void build(){
    for(M=1; M<=n+1; M<<=1);
    REP(i, 1, n) s[i+M]=a[i];
    REPd(i, M-1, 1) s[i]=f(s[2*i], s[2*i+1]);
}

ll qrange(int l, int r, ll init){ // 根据 f 传 init
    ll res=init;
    for(l=l+M-1, r=r+M+1; l^r^1; l>>=1, r>>=1){
        if(~l&1) res=f(res, s[l^1]);
        if(r&1) res=f(res, s[r^1]);
    }
    return res;
}

void edit(int x, ll v){
    for(s[x+=M]=v, x>>=1; x; x>>=1){
        s[x]=f(s[2*x], s[2*x+1]);
    }
}

ll qpoint(int x){
    return s[x+M];
}
```

### 珂朵莉树

```

struct node{
    int l, r;
    mutable int v;
    bool operator<(const node& rhs) const { return l<rhs.l; }
};

set<node> odt;
typedef set<node>::iterator iter;

iter split(ll p){
    iter tmp=odt.lower_bound((node){p, 0, 0});
    if(tmp!=odt.end()&&tmp->l==p) return tmp;
    --tmp;
    int tl=tmp->l, tr=tmp->r, tv=tmp->v;
    odt.erase(tmp);
    odt.insert((node){tl, p-1, tv});
    return odt.insert((node){p, tr, tv}).first;
}

// 【修改 & 查询】注意 split 顺序
// iter itr=split(r+1), itl=split(l);

```

## 数学

### 快速幂

```

const ll MOD=998244353; // 改模数

ll qpow(ll a, ll x){
    ll res=1;
    a%=MOD;
    while(x){
        if(x&1) res=res*a%MOD;
        a=a*a%MOD, x>>=1;
    }
    return res;
}

ll inv(ll x){ return qpow(x, MOD-2); } // 模数为质数时

```

### 高斯消元

```

const int N=110;
ll n;

```

```

double a[N][N], b[N];
void work(){
    n=rd();
    REP(i, 1, n){
        REP(j, 1, n) a[i][j]=rd();
        b[i]=rd();
    }
    REP(i, 1, n){
        int t=i;
        REP(j, i+1, n) if(abs(a[j][i])>1e-7&&(abs(a[t][i])>abs(a[j]
[i])||abs(a[t][i])<1e-7)) t=j;
        REP(j, i, n) swap(a[t][j], a[i][j]);
        if(abs(a[i][i])<1e-7){
            puts("No Solution");
            return 0;
        }
        swap(b[t], b[i]);
        double e=a[i][i];
        REP(j, i, n) a[i][j]/=e;
        b[i]/=e;
        REP(j, i+1, n){
            double d=a[j][i];
            REP(k, i, n) a[j][k]-=d*a[i][k];
            b[j]-=d*b[i];
        }
    }
    REPd(i, n, 1) REP(j, 1, i-1) b[j]-=a[j][i]*b[i], a[j][i]=0;
    // REP(i, 1, n) printf("%.2f\n", b[i]);
    // b[1...n] 保存 Ax=b 的解
}

```