# 模板合集

HocRiser 2020.10.20

测试………………………………………………………………3

莫比乌斯…………………………………………………………5

杜教筛……………………………………………………………6

Pollard-Rho………………………………………………………8

exCRT……………………………………………………………10

exLucas…………………………………………………………11

(ex)BSGS…………………………………………………………13

FFT&NTT…………………………………………………………14

FWT………………………………………………………………19

Manacher…………………………………………………………20

KMP………………………………………………………………21

AC自动机…………………………………………………………22

后缀数组…………………………………………………………25

后缀自动机………………………………………………………26

连通性……………………………………………………………28

2-SAT……………………………………………………………34

网络流……………………………………………………………35

匈牙利……………………………………………………………40

KM算法…………………………………………………………41

最小树形图………………………………………………………42

计算几何…………………………………………………………44

最小圆覆盖………………………………………………………46

半平面交…………………………………………………………48

旋转卡壳…………………………………………………………51

点分治……………………………………………………………52

可持久化Trie……………………………………………………54

Splay………………………………………………………………55

LCT………………………………………………………………57

无旋Treap………………………………………………………59

虚树………………………………………………………………61

左偏树……………………………………………………………63

块状………………………………………………………………64

树剖………………………………………………………………68

KD-Tree…………………………………………………………71

树套树……………………………………………………………73

主席树……………………………………………………………76

CDQ分治…………………………………………………………80

整体二分…………………………………………………………81

测试：

//1:

for i in $(seq 1 10); do

echo ----------&i------------

cp a$i.in a.in

time ./a

if !(diff -w a.out a$i.ans); then echo WA!!!; exit; fi

read

done

//2:

g++ std.cpp -o std -g -Wall

g++ brute.cpp -o brute -g -Wall

fpc -g make

while true; do

echo -----------------

./make

time ./std<std.in>std.out

time ./brute<std.in>brute.out

if !(diff -w std.out brute.out); then echo WA!!!; exit; fi

read

done

//3:

int main(int argc,char\* argv[]){}

//argc非传入参数，它存储一共传入了几个字符串参数。

//e.g. time ./sy 1 3 5 7

//则argc=5,argv[0]="./sy",argv[1]="1",argv[2]="3",argv[3]="5",argv[4]="7".

//4:

#include<sys/timeb.h>

void insd(){

struct timeb tp; ftime(&tp);

srand(tp.time\*1000+tp.millitm);

}

int sj(int l,int r){ return rand()%(r-l+1)+l; }

int main(){

freopen("a.in","w",stdout);

insd(); int n=sj(1,100); printf("%d\n",n);

return 0;

}

//5:

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

int main(){

system("g++ make.cpp -o make -g -Wall");

system("g++ a.cpp -o a -g -Wall");

system("g++ b.cpp -o b -g -Wall");

do{

system("make.exe");

system("a.exe");

system("b.exe");

}while (!system("fc a.out b.out"));

return 0;

}

//6:

@echo off

g++ a.cpp -o a -g -Wall -std=c++11

g++ a2.cpp -o a2 -g -Wall -std=c++11

g++ make.cpp -o make -g -Wall -std=c++11

:loop

make.exe

a.exe

a2.exe

fc a.out a2.out

if not errorlevel 1 goto loop

echo WA!!!

pause

莫比乌斯：

//O(N+Q\*sqrt(N))

LL solve(int n,int m){

LL res=0;

if (n>m) swap(n,m);

for (int i=1,l=0; i<=n; i=l+1){

l=min(n/(n/i),m/(m/i));

res+=1ll\*(sum[l]-sum[i-1])\*(n/i)\*(m/i);

}

return res;

}

int main(){

Mobius();

for (scanf("%d",&T); T--; ){

scanf("%d%d%d%d%d",&a,&b,&c,&d,&k);

LL ans=solve(b/k,d/k)-solve((a-1)/k,d/k)-solve(b/k,(c-1)/k)+solve((a-1)/k,(c-1)/k);

printf("%lld\n",ans);

}

return 0;

}

杜教筛：

//O(Q\*N^3/4),预处理O(Q\*N^2/3)

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long ll;

using namespace std;

const int N=2000100,M=100100;

int cas,n,m,cnt,c[N];

ll phi[N],mu[N],p[M],q[M];

bool vis[M];

ll get\_p(int x){ return (x<=m) ? phi[x] : p[n/x]; }

ll get\_q(int x){ return (x<=m) ? mu[x] : q[n/x]; }

void solve(int x){

if (x<=m) return;

int i,j=1,t=n/x;

if (vis[t]) return;

vis[t]=1; p[t]=((ll)x+1)\*x>>1; q[t]=1;

while (j<x){

i=j+1; j=x/(x/i); solve(x/i);

p[t]-=get\_p(x/i)\*(j-i+1); q[t]-=get\_q(x/i)\*(j-i+1);

}

}

int main(){

scanf("%d",&cas); m=2000000;

int i,j; phi[1]=mu[1]=1;

for (i=2; i<=m; i++){

if (!phi[i]) phi[i]=i-1,mu[i]=-1,c[++cnt]=i;

for (j=1; j<=cnt && i\*c[j]<=m; j++)

if (i%c[j]) phi[i\*c[j]]=phi[i]\*(c[j]-1),mu[i\*c[j]]=-mu[i];

else{ phi[i\*c[j]]=phi[i]\*c[j]; mu[i\*c[j]]=0; break; }

}

for (i=2; i<=m; i++) phi[i]+=phi[i-1],mu[i]+=mu[i-1];

while (cas--){

scanf("%d",&n); memset(vis,0,sizeof(vis));

if (n<=m) printf("%lld %lld\n",phi[n],mu[n]);

else solve(1ll\*n),printf("%lld %lld\n",p[1],q[1]);

}

return 0;

}

//记忆化

#include<cstdio>

#include<cstring>

#include<map>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long ll;

using namespace std;

const int N=5000010;

ll T,n,d,l,r,m,tot,p[N],miu[N],phi[N];

bool b[N];

map<ll,ll>mpM,mpP;

void init(ll n){

miu[1]=phi[1]=1;

rep(i,2,n){

if (!b[i]) p[++tot]=i,miu[i]=-1,phi[i]=i-1;

for (ll j=1; j<=tot && i\*p[j]<=n; j++){

b[i\*p[j]]=1;

if (i%p[j]) miu[i\*p[j]]=-miu[i],phi[i\*p[j]]=phi[i]\*(p[j]-1);

else { miu[i\*p[j]]=0; phi[i\*p[j]]=phi[i]\*p[j]; break; }

}

}

rep(i,2,n) miu[i]+=miu[i-1],phi[i]+=phi[i-1];

}

ll Miu(ll x){

if (x<=m) return miu[x];

if (mpM.count(x)) return mpM[x];

ll lst,res=1;

for (ll i=2; i<=x; i=lst+1)

lst=x/(x/i),res=res-Miu(x/i)\*(lst-i+1);

return mpM[x]=res;

}

ll Phi(ll x){

if (x<=m) return phi[x];

if (mpP.count(x)) return mpP[x];

ll lst,res=1ll\*x\*(x+1)>>1;

for (ll i=2; i<=x; i=lst+1)

lst=x/(x/i),res=res-Phi(x/i)\*(lst-i+1);

return mpP[x]=res;

}

Pollard Rho：

#include<cstdio>

#include<vector>

#include<algorithm>

#define rep(i,l,r) for (int i=(l); i<=(r); i++)

typedef long long ll;

using namespace std;

vector<ll>ls;

ll T,e,n,c,d,y,p,r,w,mx;

ll Rand(ll l,ll r){ return (((ll)rand()<<31)+rand())%(r-l+1)+l; }

ll gcd(ll a,ll b){ return (b) ? gcd(b,a%b) : a; }

ll ksc(ll &a,ll b,ll mod){

ll res=0;

for (; b; a=(a<<1)%mod,b>>=1)

if (b & 1) res=(res+a)%mod;

return res;

}

ll ksm(ll a,ll b,ll mod){

ll res=1;

for (; b; a=ksc(a,a,mod),b>>=1)

if (b & 1) res=ksc(res,a,mod);

return res;

}

ll find(ll n,int c){

ll i=1,k=2,x=Rand(0,n-1),y=x,d;

while (1){

i++; x=(ksc(x,x,n)+c)%n; d=gcd(abs(y-x),n);

if (d>1 && d<n) return d;

if (y==x) return -1;

if (i==k) y=x,k<<=1;

}

}

ll Rho(ll n,int c){ ll p=-1; while (p==-1) p=find(n,c--); return p; }

bool chk(ll a,ll n){

ll m=n-1,x,y; int k=0;

while (!(m&1)) m>>=1,k++;

x=ksm(a,m,n);

rep(i,1,k){

y=ksm(x,2,n);

if (y==1 && x!=1 && x!=n-1) return 1;

x=y;

}

return y!=1;

}

bool Miller(ll n){

if (n==2) return 1;

if (!(n&1)) return 0;

rep(i,1,3) if (chk(Rand(1,n-1),n)) return 0;

return 1;

}

void Fac(ll x,int c){

if (x==1) return;

if (Miller(x)) { ls.push\_back(x); return; }

ll p=Rho(x,c); Fac(p,c);

while (x%p==0) x/=p;

Fac(x,c);

}

int main(){

for (scanf("%lld",&T); T--; ){

scanf("%lld",&n); ls.clear();

if (Miller(n)) { puts("Prime"); continue; }

Fac(n,19260817); mx=0;

for (vector<ll>::iterator it=ls.begin(); it!=ls.end(); it++) mx=max(mx,\*it);

printf("%lld\n",mx);

}

return 0;

}

exCRT：

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (ll i=(l); i<=(r); i++)

typedef long long ll;

using namespace std;

const int N=100010;

int n;

ll a[N],b[N];

ll gcd(ll a,ll b){ return b ? gcd(b,a%b) : a; }

ll mul(ll a,ll b,ll mod){

ll res=0;

for (; b; a=(a+a)%mod,b>>=1)

if (b & 1) res=(res+a)%mod;

return res;

}

void exgcd(ll a,ll b,ll &x,ll &y){

if (!b){ x=1; y=0; return; }

exgcd(b,a%b,y,x); y-=(a/b)\*x;

}

ll inv(ll a,ll b){ ll x,y; exgcd(a,b,x,y); x=(x%b+b)%b; return x; }

void merge(ll c1,ll c2,ll m1,ll m2,ll &c3,ll &m3){

ll d=gcd(m1,m2); m3=m1/d\*m2;

if (c2<c1) swap(c1,c2),swap(m1,m2);

if ((c2-c1)%d) { puts("No solution"); exit(0); }

c3=(mul(mul(inv(m1/d,m2/d),(c2-c1)/d,m2/d),m1,m3)+c1)%m3;

}

int main(){

scanf("%d",&n);

rep(i,1,n) scanf("%lld%lld",&a[i],&b[i]);

rep(i,2,n) merge(b[i-1],b[i],a[i-1],a[i],b[i],a[i]);

printf("%lld\n",b[n]);

return 0;

}

exLucas：

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long ll;

using namespace std;

const int N=110;

ll mod,n,m,w[10],ans,x,y,Mod[N],sm[100010],st[N],r[N],num;

ll ksm(ll a,ll b,ll p){

ll res;

for (res=1; b; a=a\*a%p,b>>=1)

if (b&1) res=res\*a%p;

return res;

}

void exgcd(ll a,ll b,ll &x,ll &y){

if (!b) x=1,y=0; else exgcd(b,a%b,y,x),y-=a/b\*x;

}

ll inv(ll a,ll b){ ll x,y; exgcd(a,b,x,y); return (x%b+b)%b; }

ll F(ll n,ll pi,ll pk){

return n ? ksm(sm[pk],n/pk,pk)\*sm[n%pk]%pk\*F(n/pi,pi,pk)%pk : 1;

}

ll exlucas(ll n,ll m,ll pi,ll pk){

if (m>n) return 0;

sm[0]=sm[1]=1; rep(i,2,pk) sm[i]=sm[i-1]\*((i%pi)?i:1)%pk;

ll a=F(n,pi,pk),b=F(m,pi,pk),c=F(n-m,pi,pk),k=0;

for (ll i=n; i; i/=pi) k+=i/pi;

for (ll i=m; i; i/=pi) k-=i/pi;

for (ll i=n-m; i; i/=pi) k-=i/pi;

return a\*inv(b,pk)%pk\*inv(c,pk)%pk\*ksm(pi,k,pk)%pk;

}

ll CRT(ll n,ll r[],ll m[]){

ll res=0;

rep(i,1,n) res=(res+(mod/m[i])\*inv(mod/m[i],m[i])\*r[i])%mod;

return res;

}

ll par(ll n,ll m[],ll st[]){

ll num=0;

for (ll i=2; i\*i<=n; i++) if (n%i==0){

ll pk=1;

while (n%i==0) pk\*=i,n/=i;

m[++num]=pk; st[num]=i;

}

if (n>1) m[++num]=n,st[num]=n;

return num;

}

ll excomb(ll n,ll m){

rep(i,1,num) r[i]=exlucas(n,m,st[i],Mod[i]);

return CRT(num,r,Mod);

}

int main(){

scanf("%lld\n",&mod); scanf("%lld%lld",&n,&m);

ll sum=0; rep(i,1,m) scanf("%lld",&w[i]),sum+=w[i];

if (n<sum){ puts("Impossible"); return 0; }

num=par(mod,Mod,st); ans=1;

rep(i,1,m) n-=w[i-1],ans=ans\*excomb(n,w[i])%mod;

printf("%lld\n",ans);

return 0;

}

(EX)BSGS:

void BSGS(){

mp.clear();

if (a==0 && b==0) { puts("1"); return; }

if (b==0) { puts("no solution"); return; }

ll m=ceil(sqrt(p)); ll now=1; mp[b]=0;

rep(i,1,m) now=(now\*a)%p,mp[(b\*now)%p]=i;

ll t=pow(a,m); now=1;

rep(i,1,m){

now=(now\*t)%p;

if (mp[now]) { printf("%lld\n",i\*m-mp[now]); return; }

}

puts("no solution");

}

ll exbsgs(){

if (b==1) return 0;

ll t,d=1,k=0;

while ((t=gcd(a,p))!=1){

if (b%t) return -1;

k++; b/=t; p/=t; d=d\*(a/t)%p;

if (b==d) return k;

}

mp.clear();

ll m=ceil(sqrt(p)),now=b; t=pow(a,m);

rep(j,1,m) now=(now\*a)%p,mp[now]=j;

rep(i,1,m){

d=d\*t%p;

if (mp[d]) return i\*m-mp[d]+k;

}

return -1;

}

FFT：

#include<cstdio>

#include<complex>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

typedef complex<double> C;

const int N=262145;

const double pi=acos(-1.);

int n,m,L,x,rev[N];

C a[N],b[N];

void DFT(C a[],int f){

rep(i,0,n-1) if (i<rev[i]) swap(a[i],a[rev[i]]);

for (int i=1; i<n; i<<=1){

C wn(cos(pi/i),f\*sin(pi/i));

for (int p=i<<1,j=0; j<n; j+=p){

C w(1,0);

for (int k=0; k<i; k++,w\*=wn){

C x=a[j+k],y=w\*a[i+j+k]; a[j+k]=x+y; a[i+j+k]=x-y;

}

}

}

}

int main(){

scanf("%d%d",&n,&m);

rep(i,0,n) scanf("%d",&x),a[i]=x;

rep(i,0,m) scanf("%d",&x),b[i]=x;

m=n+m; for (n=1; n<=m; n<<=1) L++;

rep(i,0,n-1) rev[i]=(rev[i>>1]>>1)|((i&1)<<(L-1));

DFT(a,1); DFT(b,1);

rep(i,0,n-1) a[i]=a[i]\*b[i];

DFT(a,-1);

rep(i,0,m) printf("%d ",(int)(a[i].real()/n+0.5));

return 0;

}

NTT：

#include<cstdio>

#include<complex>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=262145,md=(119<<23)+1,G=3;//998244353

int n,m,L,x,rev[N],a[N],b[N];

int ksm(int a,int b){

int res;

for (res=1; b; a=(1ll\*a\*a)%md,b>>=1)

if (b & 1) res=(1ll\*res\*a)%md;

return res;

}

void FFT(int a[],int f){

rep(i,0,n-1) if (i<rev[i]) swap(a[i],a[rev[i]]);

for (int i=1; i<n; i<<=1){

int wn=ksm(G,(f==1) ? (md-1)/(i<<1) : (md-1)-(md-1)/(i<<1));

for (int p=i<<1,j=0; j<n; j+=p){

int w=1;

for (int k=0; k<i; k++,w=1ll\*w\*wn%md){

int x=a[j+k],y=1ll\*w\*a[i+j+k]%md;

a[j+k]=(x+y)%md; a[i+j+k]=(x-y+md)%md;

}

}

}

if (f==1) return;

int inv=ksm(n,md-2);

rep(i,0,n-1) a[i]=1ll\*a[i]\*inv%md;

}

int main(){

scanf("%d%d",&n,&m);

rep(i,0,n) scanf("%d",&x),a[i]=x;

rep(i,0,m) scanf("%d",&x),b[i]=x;

m=n+m; for (n=1; n<=m; n<<=1) L++;

rep(i,0,n-1) rev[i]=(rev[i>>1]>>1)|((i&1)<<(L-1));

FFT(a,1); FFT(b,1);

rep(i,0,n-1) a[i]=1ll\*a[i]\*b[i]%md;

FFT(a,-1);

rep(i,0,m) printf("%d ",a[i]);

return 0;

}

多项式运算：

#include<cmath>

#include<cstdio>

#include<algorithm>

#include<cstring>

#define mem(a) memset(a,0,sizeof(a))

#define rep(i,l,r) for (int i=(l); i<=(r); i++)

using namespace std;

const int N=500010,mod=998244353,inv2=(mod+1)/2;

int n,k,rev[N],inv[N],X[N],Y[N],A[N],B[N],C[N],D[N],E[N],F[N],G[N];

void Print(int a[],int n=::n){ for (int i=0; i<n; i++) printf("%d ",a[i]); puts(""); }

int ksm(int a,int b){

int res=1;

for (; b; a=1ll\*a\*a%mod,b>>=1)

if (b & 1) res=1ll\*res\*a%mod;

return res;

}

void NTT(int a[],int n,bool f){

for (int i=0; i<n; i++) if (i<rev[i]) swap(a[i],a[rev[i]]);

for (int i=1; i<n; i<<=1){

int wn=ksm(3,f ? (mod-1)/(i<<1) : (mod-1)-(mod-1)/(i<<1));

for (int p=i<<1,j=0; j<n; j+=p){

int w=1;

for (int k=0; k<i; k++,w=1ll\*w\*wn%mod){

int x=a[j+k],y=1ll\*w\*a[i+j+k]%mod;

a[j+k]=(x+y)%mod; a[i+j+k]=(x-y+mod)%mod;

}

}

}

if (f) return;

int inv=ksm(n,mod-2);

for (int i=0; i<n; i++) a[i]=1ll\*a[i]\*inv%mod;

}

void mul(int a[],int b[],int l){

int n=1,L=0;

for (; n<(l<<1); n<<=1) L++;

for (int i=0; i<n; i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(L-1));

NTT(a,n,1); NTT(b,n,1);

for (int i=0; i<n; i++) a[i]=1ll\*a[i]\*b[i]%mod;

NTT(a,n,0); NTT(b,n,0);

}

void Inv(int a[],int b[],int l){

if (l==1){ b[0]=ksm(a[0],mod-2); return; }

Inv(a,b,l>>1); int n=1,L=0;

for (; n<(l<<1); n<<=1) L++;

for (int i=0; i<n; i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(L-1));

for (int i=0; i<l; i++) A[i]=a[i];

NTT(A,n,1); NTT(b,n,1);

for (int i=0; i<n; i++) b[i]=1ll\*b[i]\*(2-1ll\*A[i]\*b[i]%mod+mod)%mod;

NTT(b,n,0);

for (int i=l; i<n; i++) b[i]=0;

for (int i=0; i<n; i++) A[i]=0;

}

void Sqrt(int a[],int b[],int l){

if (l==1){ b[0]=sqrt(a[0]); return; }

Sqrt(a,b,l>>1); Inv(b,B,l); int n=1,L=0;

for (; n<(l<<1); n<<=1) L++;

for (int i=0; i<n; i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(L-1));

for (int i=0; i<l; i++) C[i]=a[i];

NTT(C,n,1); NTT(B,n,1); NTT(b,n,1);

for (int i=0; i<n; i++) b[i]=1ll\*inv2\*(b[i]+1ll\*C[i]\*B[i]%mod)%mod;

NTT(b,n,0);

for (int i=l; i<n; i++) b[i]=0;

for (int i=0; i<n; i++) C[i]=B[i]=0;

}

void Deri(int a[],int b[],int l){

for (int i=1; i<l; i++) b[i-1]=1ll\*i\*a[i]%mod;

}

void Inte(int a[],int b[],int l){

for (int i=1; i<l; i++) b[i]=1ll\*a[i-1]\*inv[i]%mod; b[0]=0;

}

void Ln(int a[],int b[],int l){

Deri(a,D,l); Inv(a,E,l); mul(D,E,l); Inte(D,b,l);

for (int i=0; i<(l<<1); i++) D[i]=E[i]=0;

}

void Exp(int a[],int b[],int l){

if (l==1){ b[0]=1; return; }

Exp(a,b,l>>1); Ln(b,F,l); int n=1,L=0;

for (; n<(l<<1); n<<=1) L++;

for (int i=0; i<n; i++) rev[i]=(rev[i>>1]>>1)|((i&1)<<(L-1));

for (int i=0; i<l; i++) F[i]=(-F[i]+a[i]+mod)%mod; F[0]=(F[0]+1)%mod;

NTT(F,n,1); NTT(b,n,1);

for (int i=0; i<n; i++) b[i]=1ll\*b[i]\*F[i]%mod;

NTT(b,n,0);

for (int i=l; i<n; i++) b[i]=0;

for (int i=0; i<n; i++) F[i]=0;

}

void Ksm(int a[],int b[],int k,int l){

Ln(a,G,l);

for (int i=0; i<l; i++) G[i]=1ll\*k\*G[i]%mod;

Exp(G,b,l);

}

int main(){

scanf("%d%d",&n,&k);

for (int i=0; i<n; i++) scanf("%d",&X[i]);

int l=1; for (; l<=n; l<<=1); inv[0]=inv[1]=1;

rep(i,2,l) inv[i]=1ll\*(mod-mod/i)\*inv[mod%i]%mod;

Sqrt(X,Y,l); mem(X); //Print(Y);

Inv(Y,X,l); mem(Y); //Print(X);

Inte(X,Y,l); mem(X); //Print(Y);

Exp(Y,X,l); mem(Y); //Print(X);

Inv(X,Y,l); Y[0]=(Y[0]+1)%mod; mem(X); //Print(Y);

Ln(Y,X,l); X[0]=(X[0]+1)%mod; mem(Y); //Print(X);

Ksm(X,Y,k,l); mem(X); //Print(Y);

Deri(Y,X,n); mem(Y); Print(X);

return 0;

}

FWT：

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long ll;

using namespace std;

const int N=(1<<18)+5,mod=998244353,rev=499122177;

int n,m,len,s[N],t[N],a[N],b[N];

void FWT(int a[],int n,int f,int op){

for (int i=1; i<n; i<<=1)

for (int p=i<<1,j=0; j<n; j+=p)

for (int k=0; k<i; k++){

int x=a[j+k],y=a[i+j+k];

if (op==1){//or

if (f==1) a[i+j+k]=(x+y)%mod; else a[i+j+k]=(y-x+mod)%mod;

}

if (op==2){//and

if (f==1) a[j+k]=(x+y)%mod; else a[j+k]=(x-y+mod)%mod;

}

if (op==3){//xor

if (f==1) a[j+k]=(x+y)%mod,a[i+j+k]=(x-y+mod)%mod;

else a[j+k]=1ll\*(x+y)\*rev%mod,a[i+j+k]=((1ll\*(x-y)\*rev)%mod+mod)%mod;

}

}

}

int main(){

scanf("%d",&n); n=1<<n; len=n+n;

for (int i=0; i<n; i++) scanf("%d",&s[i]);

for (int i=0; i<n; i++) scanf("%d",&t[i]);

rep(op,1,3){

for (int i=0; i<len; i++) a[i]=s[i],b[i]=t[i];

FWT(a,len,1,op); FWT(b,len,1,op);

for (int i=0; i<len; i++) a[i]=1ll\*a[i]\*b[i]%mod;

FWT(a,len,-1,op);

for (int i=0; i<n; i++) printf("%d ",a[i]); puts("");

}

return 0;

}

Manacher：

void get(){

id=mx=mxl=0; len=strlen(s+1);

rep(i,1,len) s1[i\*2]=s[i],s1[i\*2+1]='#';

s1[0]='$'; s1[1]='#'; n=len\*2+1;

rep(i,1,n){

p[i]=(mx>i) ? min(p[2\*id-i],mx-i) : 1;

while (s1[i+p[i]]==s1[i-p[i]]) p[i]++;

if (p[i]+i>mx) mx=p[i]+i,id=i;

mxl=max(mxl,p[i]);

}

printf("%d\n",mxl-1);

}

KMP:

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=100010;

int n,p,nxt[N];

char s[N];

void getnxt(){

int j=0; nxt[0]=nxt[1]=0;

rep(i,2,n){

while (j && s[j+1]!=s[i]) j=nxt[j];

if (s[j+1]!=s[i]) nxt[i]=0; else nxt[i]=++j;

}

}

int get(){

int res=0,j=0;

rep(i,2,n){

while (j && s[j+1]!=s[i]) j=nxt[j];

if (s[j+1]==s[i]) j++;

if (j==p) res++,j=nxt[j];

}

return res;

}

int main(){

scanf("%s",s+1); n=strlen(s+1);

getnxt();

for (p=nxt[n]; p && get()<2; p=nxt[p]);

if (p==0) puts("Just a legend"); else rep(i,1,p) printf("%c",s[i]);

return 0;

}

AC自动机：

#include<cstdio>

#include<algorithm>

#include<cstring>

#define rep(i,l,r) for (register int i=l; i<=r; ++i)

using namespace std;

const int N=6010,M=110,md=10007;

int q[N],ch[N][27],f[M][N],fail[N],cnt,n,m;

char s[M],w[N];

void clear(int x){ memset(ch[x],0,sizeof(ch[x])); fail[x]=0; }

void ins(char s[]){

int l=strlen(s),x=0;

rep(i,0,l-1){

int t=s[i]-'A'+1;

if (!ch[x][t]) clear(++cnt),ch[x][t]=cnt;

x=ch[x][t];

}

w[x]=1;

}

void build(){

int st=1,ed=0;

rep(i,1,26) if (ch[0][i]) q[++ed]=ch[0][i];

while (st!=ed+1){

int x=q[st++];

rep(i,1,26)

if (ch[x][i]) q[++ed]=ch[x][i],fail[ch[x][i]]=ch[fail[x]][i];

else ch[x][i]=ch[fail[x]][i];

w[x]|=w[fail[x]];

}

}

void work(){

int ans1=1,ans2=0; f[0][0]=1;

rep(i,1,m) rep(j,0,cnt) if (!w[j])

rep(k,1,26) f[i][ch[j][k]]=(f[i][ch[j][k]]+f[i-1][j])%md;

rep(i,0,cnt) if (!w[i]) ans2+=f[m][i];

rep(i,1,m) ans1=ans1\*26%md;

printf("%d\n",((ans1-ans2)%md+md)%md);

}

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (register int i=l; i<=r; ++i)

#define For(i,x) for (int i=h[x]; i; i=e[i].nxt)

using namespace std;

const int N=100100;

int cnt,tim,tot,now,len,k,nd,st,ed,m,n;

int ch[N][27],c[N],h[N],l[N],r[N],fa[N],pos[N],q[N],fail[N],ans[N];

char s[N];

struct P{ int x,y,id; }a[N];

struct E{ int to,nxt; }e[N];

void add(int u,int v){ e[++cnt]=(E){v,h[u]}; h[u]=cnt; }

void ins(int x,int k){ for (int i=x; i<=tim; i+=i & -i) c[i]+=k; }

int que(int x){ int res=0; for (int i=x; i; i-=i & -i) res+=c[i]; return res; }

void dfs(int x){

l[x]=r[x]=++tim;

For(i,x) dfs(e[i].to),r[x]=r[e[i].to];

}

void work(){

int t=0; now=0; k=1;

rep(i,1,len){

if (s[i]=='B') { ins(l[now],-1); now=fa[now]; continue; }

if (s[i]=='P') {

for (t++; a[k].y==t && k<=m; k++)

ans[a[k].id]=que(r[pos[a[k].x]])-que(l[pos[a[k].x]]-1); continue;

}

now=ch[now][s[i]-'a']; ins(l[now],1);

}

}

int cmp(P a,P b){ return a.y<b.y; }

int main(){

scanf("%s",s+1); len=strlen(s+1); now=0;

rep(i,1,len){

if (s[i]=='B') { now=fa[now]; continue; }

if (s[i]=='P') { pos[++tot]=now; continue; }

int t=s[i]-'a';

if (ch[now][t]) now=ch[now][t]; else ch[now][t]=++nd,fa[nd]=now,now=nd;

}

st=1; ed=0;

rep(i,0,25) if (ch[0][i]) q[++ed]=ch[0][i];

while (st!=ed+1){

int x=q[st++];

rep(i,0,25)

if (ch[x][i]) q[++ed]=ch[x][i],fail[ch[x][i]]=ch[fail[x]][i];

else ch[x][i]=ch[fail[x]][i];

}

rep(i,1,nd) add(fail[i],i);

dfs(0); scanf("%d",&m);

rep(i,1,m) scanf("%d%d",&a[i].x,&a[i].y),a[i].id=i;

sort(a+1,a+m+1,cmp); work();

rep(i,1,m) printf("%d\n",ans[i]);

return 0;

}

后缀数组：

#include<cstdio>

#include<cstring>

#include<algorithm>

#define LCP(x,y) SA.que(x,y)

#define LCS(x,y) SA1.que(n-y+1,n-x+1)

#define rep(i,l,r) for (int i=l; i<=r; i++)

const int N=30010;

typedef long long LL;

using namespace std;

int n,T,log[N],f[N],g[N];

char s[N]; LL ans;

struct sa\_array{

int sa[N],rk[N],h[N],st[N][16],x[N],y[N],c[N]; char S[N];

int Cmp(int a,int b,int l){ return y[a]==y[b] && y[a+l]==y[b+l]; }

void build\_sa(int m){

memset(y,0,sizeof(y));

rep(i,0,m) c[i]=0;

rep(i,1,n) c[x[i]=S[i]-'a'+1]++;

rep(i,1,m) c[i]+=c[i-1];

for (int i=n; i; i--) sa[c[x[i]]--]=i;

for (int k=1,p=0; p<n; k<<=1,m=p){

p=0;

rep(i,n-k+1,n) y[++p]=i;

rep(i,1,n) if (sa[i]>k) y[++p]=sa[i]-k;

rep(i,0,m) c[i]=0;

rep(i,1,n) c[x[y[i]]]++;

rep(i,1,m) c[i]+=c[i-1];

for (int i=n; i; i--) sa[c[x[y[i]]]--]=y[i];

rep(i,1,n) y[i]=x[i]; p=1; x[sa[1]]=1;

rep(i,2,n) x[sa[i]]=Cmp(sa[i-1],sa[i],k) ? p : ++p;

}

}

void get(){

int k=0;

rep(i,1,n) rk[sa[i]]=i;

rep(i,1,n){

for (int j=sa[rk[i]-1]; i+k<=n && j+k<=n && S[i+k]==S[j+k]; k++);

h[rk[i]]=k; if (k) k--;

}

}

void rmq(){

rep(i,1,n) st[i][0]=h[i];

rep(i,1,log[n])

rep(j,1,n-(1<<i)+1) st[j][i]=min(st[j][i-1],st[j+(1<<(i-1))][i-1]);

}

int ask(int l,int r){

l++; int t=log[r-l+1];

return min(st[l][t],st[r-(1<<t)+1][t]);

}

int que(int x,int y){ return ask(min(rk[x],rk[y]),max(rk[x],rk[y]));}

}SA,SA1;

void solve(){

memset(f,0,sizeof(f)); memset(g,0,sizeof(g));

for (int len=1,x,y,l,r; 2\*len<=n; len++)

for (int i=1,j=len+1; j<=n; i+=len,j+=len)

if (s[i]==s[j]){

x=LCS(i,j); y=LCP(i,j);

l=max(i,i-x+len); r=min(i+y,j);

if (r>l){

f[l+len]++; f[r+len]--;

g[l-len+1]++; g[r-len+1]--;

}

}

rep(i,2,n) f[i]+=f[i-1],g[i]+=g[i-1];

rep(i,1,n-1) ans+=(LL)f[i]\*g[i+1];

}

int main(){

log[1]=0; rep(i,2,N) log[i]=log[i>>1]+1;

scanf("%d",&T);

while (T--){

scanf("%s",s+1); n=strlen(s+1);

rep(i,1,n) SA.S[i]=SA1.S[n-i+1]=s[i];

SA.build\_sa(30); SA.get(); SA.rmq();

SA1.build\_sa(30); SA1.get(); SA1.rmq();

ans=0; solve(); printf("%lld\n",ans);

}

return 0;

}

后缀自动机：

//两个节点所代表的right集合若非包含关系则不相交

//一个节点所代表的最短串长度mn[x]=mx[fa[x]]+1

//走son即左端集合不变（可能增加），最大被包含，走fa即右端集合不变，最小包含。

//sigma(y|fa[y]==x)即x的右端点集合完全合并。

#include<cstdio>

#include<cstring>

#include<algorithm>

#define mem(a) memset(a,0,sizeof(a))

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=40010;

int n,lst,cnt,ans,l[N],r[N],mx[N],fa[N],a[N],son[N][180],q[N],v[N];

void init(){

memset(l,127,sizeof(l));

mem(r); mem(v); mem(mx); mem(fa); mem(son);

lst=cnt=1; ans=0;

}

void ext(int c){

int p=lst,np=lst=++cnt; mx[np]=mx[p]+1;

l[np]=r[np]=mx[np];

while (!son[p][c] && p) son[p][c]=np,p=fa[p];

if (!p) fa[np]=1;

else{

int q=son[p][c];

if (mx[p]+1==mx[q]) fa[np]=q;

else{

int nq=++cnt; mx[nq]=mx[p]+1;

memcpy(son[nq],son[q],sizeof(son[q]));

fa[nq]=fa[q]; fa[np]=fa[q]=nq;

while (son[p][c]==q) son[p][c]=nq,p=fa[p];

}

}

}

void solve(){

rep(i,1,cnt) v[mx[i]]++;

rep(i,1,n) v[i]+=v[i-1];

for (int i=cnt; i; i--) q[v[mx[i]]--]=i;

for (int i=cnt; i; i--){

int p=q[i];

l[fa[p]]=min(l[fa[p]],l[p]); r[fa[p]]=max(r[fa[p]],r[p]);

}

rep(i,1,cnt) ans=max(ans,min(mx[i],r[i]-l[i]));

if (ans<4) puts("0"); else printf("%d\n",ans+1);

}

int main(){

while (scanf("%d",&n)){

if (!n) break;

rep(i,1,n) scanf("%d",a+i); n--;

rep(i,1,n) a[i]=a[i+1]-a[i]+88;

init(); rep(i,1,n) ext(a[i]); solve();

}

return 0;

}

连通性：

//割点与桥:

var

i,m,n,x,y,time,cnt:longint;

a,g,ne,dfn,low,father:array[0..10000]of longint;

b:array[0..10000]of boolean;

function min(x,y:longint):longint;

begin

if x<y then exit(x) else exit(y);

end;

procedure add(x,y:longint);

begin

inc(cnt); g[cnt]:=y; ne[cnt]:=a[x]; a[x]:=cnt;

end;

procedure tarjan(u,fa:longint);

var i,v:longint;

begin

father[u]:=fa; inc(time);

low[u]:=time; dfn[u]:=time; i:=a[u];

while i<>0 do

begin

v:=g[i];

if dfn[v]=0 then

begin

tarjan(v,u); low[u]:=min(low[u],low[v]);

end

else

if fa<>v then low[u]:=min(low[u],dfn[v]);

i:=ne[i];

end;

end;

procedure count;

var sum,i,v:longint;

begin

tarjan(1,0); sum:=0;

for i:=2 to n do

begin

v:=father[i];

if v=1 then inc(sum)

else

if dfn[v]<=low[i] then b[v]:=true;

end;

if sum>1 then b[1]:=true;

for i:=1 to n do

if b[i] then write(i,' '); writeln;

for i:=2 to n do

begin

v:=father[i];

if dfn[v]<low[i] then writeln(v,',',i);

end;

end;

begin

readln(n,m); time:=0;

for i:=1 to m do

begin

readln(x,y); add(x,y); add(y,x);

end;

count;

end.

//点DCC:

var

time,top,cnt,x,y,n,m,i,e:longint;

dfn,low,stack,father,g,ne,a:array[0..10000]of longint;

procedure add(x,y:longint);

begin

inc(e); g[e]:=y; ne[e]:=a[x]; a[x]:=e;

end;

procedure dfs(u:longint);

var i,v,k:longint;

begin

inc(time); dfn[u]:=time; low[u]:=time;

inc(top); stack[top]:=u; i:=a[u];

while i<>0 do

begin

v:=g[i];

if dfn[v]=0 then

begin

father[v]:=u; dfs(v);

if low[v]<low[u] then low[u]:=low[v];

if low[v]>=dfn[u] then

begin

inc(cnt); write(cnt,' ');

repeat

k:=stack[top]; dec(top); write(' ',k);

until v=k;

writeln(' ',u);

end;

end

else

if (v<>father[u])and(dfn[v]<low[u]) then low[u]:=dfn[v];

i:=ne[i];

end;

end;

begin

assign(input,'DCC.in'); reset(input);

assign(output,'DCC.out'); rewrite(output);

readln(n,m); cnt:=0; time:=0; top:=0; e:=0;

for i:=1 to m do

begin

readln(x,y); add(x,y); add(y,x);

end;

dfs(1); close(input); close(output);

end.

//边DCC:

var

i,m,n,x,y,time,cnt,sum:longint;

a,g,ne,dfn,low,id:array[0..10000]of longint;

b:array[0..10000]of boolean;

function min(x,y:longint):longint;

begin

if x<y then exit(x) else exit(y);

end;

procedure add(x,y:longint);

begin

inc(cnt); g[cnt]:=y; ne[cnt]:=a[x]; a[x]:=cnt;

end;

procedure tarjan(u,fa:longint);

var i,v:longint;

begin

inc(time);

low[u]:=time; dfn[u]:=time; i:=a[u];

while i<>0 do

begin

v:=g[i];

if dfn[v]=0 then

begin

tarjan(v,u); low[u]:=min(low[u],low[v]);

if dfn[u]<low[v] then b[i]:=false;

end

else

if fa<>v then low[u]:=min(low[u],dfn[v]);

i:=ne[i];

end;

end;

procedure dfs(x:longint);

var i,k:longint;

begin

id[x]:=sum; write(' ',x); i:=a[x];

while i<>0 do

begin

k:=g[i]; if (b[i])and(id[k]=0) then dfs(k); i:=ne[i];

end;

end;

procedure work;

var i:longint;

begin

tarjan(1,0); sum:=0;

for i:=1 to n do

if id[i]=0 then

begin inc(sum); write(sum); dfs(i); writeln; end;

end;

begin

assign(input,'edgeDCC.in'); reset(input);

assign(output,'edgeDCC.out'); rewrite(output);

readln(n,m); time:=0; fillchar(b,sizeof(b),true);

for i:=1 to m do

begin

readln(x,y); add(x,y); add(y,x);

end;

work;

close(input); close(output);

end.

//tarjan:

var

i,m,n,cnt,top,tot,x,y:longint;

low,time,stack,ne,a,g,belong:array[0..1000]of longint;

instack:array[0..1000]of boolean;

procedure tarjan(x:longint);

var i,k:longint;

begin

inc(cnt); time[x]:=cnt; low[x]:=cnt;

instack[x]:=true; inc(top); stack[top]:=x; i:=a[x];

while i<>0 do

begin

k:=g[i];

if time[k]=0 then

begin

tarjan(k);

if low[k]<low[x] then low[x]:=low[k];

end

else

if (instack[k])and(time[k]<low[x]) then low[x]:=time[k];

i:=ne[i];

end;

if time[x]=low[x] then

begin

inc(tot);

repeat

i:=stack[top]; dec(top);

instack[i]:=false;

belong[i]:=tot;

until i=x;

end;

end;

void tarjan(int u){

low[u]=dfn[u]=++times;

sta[++top]=u;

for(int i=head[u];~i;i=e[i].next){

int v=e[i].to;

if(!dfn[v]){ tarjan(v); low[u]=min(low[u],low[v]); }

else if(!bleg[v]) low[u]=min(low[u],dfn[v]);

}

if(dfn[u]==low[u]){

bleg[0]++; cnt[bleg[0]]=0;

do{

bleg[sta[top]]=bleg[0];

cnt[bleg[0]]+=(sta[top]<=ncnt);

}while(sta[top--]!=u);

}

}

begin

readln(n,m); top:=0; tot:=0; cnt:=0;

fillchar(a,sizeof(a),0); fillchar(time,sizeof(time),0);

for i:=1 to m do

begin

readln(x,y); g[i]:=y; ne[i]:=a[x]; a[x]:=i;

end;

for i:=1 to n do

if time[i]=0 then tarjan(i);

writeln(i);

for i:=1 to n do

writeln('The ',i,'th node belongs to the SCC',belong[i]);

end.

2-SAT:

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

#define For(i,x) for (int i=h[x],k; i; i=e[i].nxt)

using namespace std;

const int N=1010;

char ch1,ch2;

int T,x,y,xp,yp,f,n,m,cnt,tim,scc,top,h[N],dfn[N],low[N],q[N],bel[N],inq[N];

struct E{ int to,nxt;} e[20100];

void add(int u,int v){ e[++cnt]=(E){v,h[u]}; h[u]=cnt; }

void tarjan(int x){

low[x]=dfn[x]=++tim; inq[x]=1; q[++top]=x;

For(i,x)

if (!dfn[k=e[i].to]) tarjan(k),low[x]=min(low[x],low[k]);

else if (inq[k]) low[x]=min(low[x],dfn[e[i].to]);

if (low[x]==dfn[x]){

scc++; int t;

do{ t=q[top--]; bel[t]=scc; inq[t]=0; } while (t!=x);

}

}

int main(){

for (scanf("%d",&T); T--; ){

top=tim=cnt=scc=f=0; scanf("%d%d",&n,&m);

rep(i,1,2\*n) h[i]=dfn[i]=0;

rep(i,1,m){

scanf(" %c%d %c%d",&ch1,&x,&ch2,&y);

x=x\*2-(ch1=='h'); y=y\*2-(ch2=='h');

if (x&1) xp=x++; else xp=x--;

if (y&1) yp=y++; else yp=y--;

add(xp,y); add(yp,x);

}

rep(i,1,2\*n) if (!dfn[i]) tarjan(i);

rep(i,1,n) if (bel[2\*i]==bel[2\*i-1]) { f=1; break; }

if (!f) puts("GOOD"); else puts("BAD");

}

return 0;

}

网络流：

//Dinic(O(V2\*E),n<=100000,m<=500000),POJ1149:

#include<cstdio>

#include<cstring>

#include<algorithm>

#include<vector>

#define rep(i,l,r) for (int i=l; i<=r; i++)

#define For(i,x) for (int i=h[x],k; i; i=e[i].nxt)

using namespace std;

const int N=105,inf=1000000000;

vector<int> a[N];

int v,x,y,m,n,S,T,cnt=1,d[N],q[N],h[N],cur[N],pig[1005],L[1005];

struct E{ int to,nxt,v; }e[100005];

void add(int u,int v,int w){

e[++cnt]=(E){v,h[u],w}; h[u]=cnt;

e[++cnt]=(E){u,h[v],0}; h[v]=cnt;

}

bool bfs(){

memset(d,0,sizeof(d)); q[1]=S; d[S]=1;

for (int st=0,ed=1; st!=ed; ){

int x=q[++st];

For(i,x) if (e[i].v && !d[k=e[i].to])

d[k]=d[x]+1,q[++ed]=k;

}

return d[T];

}

int dfs(int x,int lim){

if (x==T) return lim;

int t,c=0;

for (int i=cur[x],k; i; i=e[i].nxt)

if (d[k=e[i].to]==d[x]+1){

t=dfs(k,min(lim-c,e[i].v));

e[i].v-=t; e[i^1].v+=t; c+=t;

if (!e[i].v) cur[x]=i;

if (c==lim) return lim;

}

if (!c) d[x]=-1; return c;

}

int dinic(){

int ans=0;

while (bfs()){

rep(i,0,T) cur[i]=h[i];

ans+=dfs(0,inf);

}

return ans;

}

int main(){

scanf("%d%d",&m,&n); S=0; T=n+1;

rep(i,1,m) scanf("%d",pig+i);

rep(i,1,n){

for (scanf("%d",&x); x--; ) scanf("%d",&y),a[i].push\_back(y);

scanf("%d",&y); add(i,T,y);

}

rep(i,1,n)

rep(j,0,(int)a[i].size()-1){

v=a[i][j];

if(!L[v]) add(0,i,pig[v]); else add(L[v],i,inf);

L[v]=i;

}

printf("%d\n",dinic());

return 0;

}

最小费用最大流(MCMF,O(KV\*E2),n<=5000,m<=10000)

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=2010,inf=100000000;

int n,m,S,T,ans,u,v,c,st,ed,mn,cnt,h[N],pre[N],inq[N\*100],dis[N],q[N];

struct E{ int to,c,nxt,f; }e[50100];

void add(int u,int v,int w,int f){

e[++cnt]=(E){v,w,h[u],f}; h[u]=cnt;

e[++cnt]=(E){u,-w,h[v],0}; h[v]=cnt;

}

int spfa(){

rep(i,0,n+2) pre[i]=-1,inq[i]=0,dis[i]=inf;

inq[S]=1; dis[S]=0; st=0; ed=1; q[1]=S;

while (st!=ed){

int x=q[++st]; inq[x]=0;

for (int i=h[x],k; i; i=e[i].nxt)

if (e[i].f && dis[k=e[i].to]>dis[x]+e[i].c){

dis[k]=dis[x]+e[i].c; pre[k]=i;

if (!inq[k]) inq[k]=1,q[++ed]=k;

}

}

return dis[T]!=dis[0];

}

void mcmf(){

for (ans=0; spfa(); ans+=dis[T]\*mn){

mn=inf;

for (int i=pre[T]; ~i; i=pre[e[i^1].to]) mn=min(mn,e[i].f);

for (int i=pre[T]; ~i; i=pre[e[i^1].to]) e[i].f-=mn,e[i^1].f+=mn;

}

}

int main(){

while (scanf("%d%d",&n,&m)==2){

memset(h,0,sizeof(h)); cnt=1; S=n+1; T=n+2;

rep(i,1,m) scanf("%d%d%d",&u,&v,&c),add(u,v,c,1),add(v,u,c,1);

add(S,1,0,2); add(n,T,0,2); mcmf(); printf("%d\n",ans);

}

return 0;

}

上下界网络流：

1.无源汇可行流

增设超级源汇，设L[i]为上界，R[i]为下界，D[u]为∑L[v](<v,u>∈E)−∑R[v](<u,v>∈E)。

对于每条边，连R[i]-L[i]的边。

对于每个点，若D[u]为负，则从u连−D[u]到超级汇，否则从超级源连D[u]到u。

rep(i,1,m){

u=rd(); v=rd(); low=rd(); up=rd();

add(u,v,up-low); d[v]+=low; d[u]-=low; fl[i]=low;

}

rep(i,1,n) if (d[i]>0) add(S,i,d[i]),sm+=d[i]; else add(i,T,-d[i]);

ans=dinic(); if (ans<sm) { puts("NO"); return 0; }

puts("YES");

rep(i,1,m) printf("%d\n",f[(i<<1)|1]+fl[i]);

2.有源汇可行流

从汇到源连容量无穷的无下界的边，再跑无源汇可行流即可。

3.有源汇最大流

先跑可行流，删去超级源汇后跑最大流。（实际上不需要删去，下同）

跑一遍S-T有源汇可行流，再跑s-t最大流，第二次的结果就是答案。

n=rd(); m=rd(); s=rd(); t=rd(); S=n+1; T=n+2; cnt=1;

rep(i,1,m){

u=rd(); v=rd(); low=rd(); up=rd();

add(u,v,up-low); d[v]+=low; d[u]-=low; fl[i]=low;

}

rep(i,1,n) if (d[i]>0) add(S,i,d[i]),sm+=d[i]; else add(i,T,-d[i]);

add(t,s,inf); ans=dinic();

if (ans<sm) { puts("please go home to sleep"); return 0; }

S=s; T=t; printf("%d\n",dinic());

4.有源汇最小流

法一：先不连汇到源的边跑一次超级源汇的最大流，连了汇到源的边之后再跑一次超级源汇的最大流，答案即为边t->s,inf的实际流量.

法二：先连源汇跑可行流，删去t->s边后，再在残量网络上跑t->s最大流，相减得答案。

法三：考虑到法二第二次求出的新增流多算了t->s边上的流，且如果不删t->s边的话，这条边必定被跑满（即inf-可行流），故最小流=可行流-(新增流-(inf-可行流))=inf-新增流。

（和最大流只有最后一行不一样）

n=rd(); m=rd(); s=rd(); t=rd(); S=n+1; T=n+2; cnt=1;

rep(i,1,m){

u=rd(); v=rd(); low=rd(); up=rd();

add(u,v,up-low); d[v]+=low; d[u]-=low; fl[i]=low;

}

rep(i,1,n) if (d[i]>0) add(S,i,d[i]),sm+=d[i]; else add(i,T,-d[i]);

add(t,s,inf); ans=dinic();

if (ans<sm) { puts("please go home to sleep"); return 0; }

S=t; T=s; printf("%d\n",inf-dinic());

匈牙利:

var

i,x,y,e,n,s:longint;

map:array[1..1010,1..1010]of boolean;//记录读入的边

mark:array[1..1010]of boolean;//标记在一个点是否在增广路中

lk:array[1..1010]of longint;//不是记录增广路，而是记录匹配边

function find(x:longint):boolean;

var i:longint;

begin

for i:=1 to n do

if (map[x,i])and(not(mark[i])) then

begin

mark[i]:=true;

if (lk[i]=0)or(find(lk[i])) then

begin lk[i]:=x; exit(true); end;

end;

exit(false);//表示找不到合适的边，该增广路不合理。

end;

begin

fillchar(map,sizeof(map),0);

readln(n,e);

for i:=1 to e do

begin

readln(x,y); map[x,y]:=true;//邻接表储存

end;

s:=0;

for i:=1 to n do

begin

fillchar(mark,sizeof(mark),false);

if find(i) then inc(s);

end;

writeln(s);

end.

KM算法：

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=310,inf=0x3f3f3f3f;

int n,lk[N],lx[N],ly[N],vx[N],vy[N],s[N],w[N][N];

int dfs(int x){

vx[x]=1;

rep(y,1,n){

if (vy[y]) continue;

int t=lx[x]+ly[y]-w[x][y];

if (t==0){

vy[y]=1;

if (lk[y]==-1 || dfs(lk[y])) { lk[y]=x; return 1;}

}else s[y]=min(s[y],t);

}

return 0;

}

int KM(){

memset(lk,-1,sizeof(lk));

memset(lx,-inf,sizeof(lx));

memset(ly,0,sizeof(ly));

rep(i,1,n) rep(j,1,n) if (w[i][j]>lx[i]) lx[i]=w[i][j];

rep(x,1,n){

rep(i,1,n) s[i]=inf;

while (1){

memset(vx,0,sizeof(vx));

memset(vy,0,sizeof(vy));

if (dfs(x)) break;

int d=inf;

rep(i,1,n) if (!vy[i] && d>s[i]) d=s[i];

rep(i,1,n) if (vx[i]) lx[i]-=d;

rep(i,1,n) if (vy[i]) ly[i]+=d; else s[i]-=d;

}

}

int res=0;

rep(i,1,n) if (lk[i]!=-1) res+=w[lk[i]][i];

return res;

}

最小树形图：

const inf=10000000;

type

rec=record

u,v,w:longint;

end;

var

i,m,n,ans,ansi,sum:longint;

e:array[0..1000100]of rec;

ind,vis,id,pre:array[0..1010]of longint;

function zhuliu(rt,n,m:longint):longint;

var u,v,i,cnt,tn:longint;

begin

cnt:=0;

while true do

begin

tn:=0;

for i:=1 to n do ind[i]:=inf;

for i:=1 to m do

begin

u:=e[i].u; v:=e[i].v;

if (e[i].w<ind[v])and(u<>v) then

begin

pre[v]:=u; ind[v]:=e[i].w;

if u=rt then ansi:=i;

end;

end;

for i:=1 to n do

if (i<>rt)and(ind[i]=inf) then exit(-1);

for i:=1 to n do

begin vis[i]:=-1; id[i]:=-1; end;

ind[rt]:=0;

for i:=1 to n do

begin

inc(cnt,ind[i]); v:=i;

while (vis[v]<>i)and(id[v]=-1)and(v<>rt) do

begin vis[v]:=i; v:=pre[v]; end;

if (v<>rt)and(id[v]=-1) then

begin

u:=pre[v]; inc(tn);

while u<>v do

begin id[u]:=tn; u:=pre[u]; end;

id[v]:=tn;

end;

end;

if tn=0 then break;

for i:=1 to n do

if id[i]=-1 then

begin inc(tn); id[i]:=tn; end;

for i:=1 to m do

begin

v:=e[i].v;

e[i].u:=id[e[i].u]; e[i].v:=id[e[i].v];

if e[i].u<>e[i].v then dec(e[i].w,ind[v]);

end;

n:=tn; rt:=id[rt];

end;

exit(cnt);

end;

begin

while not(eof) do

begin

sum:=0; readln(n,m);

for i:=1 to m do

begin

readln(e[i].u,e[i].v,e[i].w);

inc(e[i].u); inc(e[i].v); inc(sum,e[i].w);

end;

inc(sum);

for i:=m+1 to m+n do

begin

e[i].u:=n+1; e[i].v:=i-m; e[i].w:=sum;

end;

ans:=zhuliu(n+1,n+1,n+m);

if (ans=-1)or(ans>=2\*sum) then writeln('impossible')

else writeln(ans-sum,' ',ansi-m-1);

writeln; readln;

end;

end.

计算几何：

#include<cstdio>

#include<algorithm>

#include<cmath>

#define abs(x) ((x)>0 ? (x) : -(x))

using namespace std;

const double eps=1e-8,PI=acos(-1.0);//3.14159265358979323846

struct point{ double x,y; };

struct line{ point a,b; };

point operator - (point a,point b){ return (point){a.x-b.x,a.y-b.y}; }

double dist2(point a,point b){ point t=a-b; return t.x\*t.x+t.y\*t.y; }

int zero(double x) { return (x>0 ? x : -x)<eps; }

int sign(double x) { return (x>eps) ? 1 : ((x<-eps) ? -1 : 0); }

double xmult(point p0,point p1,point p2)

{ return (p1.x-p0.x)\*(p2.y-p0.y)-(p2.x-p0.x)\*(p1.y-p0.y); }

double dmult(point p0,point p1,point p2)

{ return (p1.x-p0.x)\*(p2.x-p0.x)+(p1.y-p0.y)\*(p2.y-p0.y); }

double dist(point p1,point p2)

{ return sqrt((p1.x-p2.x)\*(p1.x-p2.x)+(p1.y-p2.y)\*(p1.y-p2.y)); }

int dotinline (point p1,point p2,point p3){ return zero(xmult(p1,p2,p3)); }//判三点共线

int dot\_online\_in(point p,line l)//判点是否在线段上,包括端点

{ return dotinline(p,l.a,l.b) && (l.a.x-p.x)\*(l.b.x-p.x)<eps && (l.a.y-p.y)\*(l.b.y-p.y)<eps; }

int dot\_online\_ex(point p,line l)//判点是否在线段上,不包括端点

{ return dot\_online\_in(p,l) && !zero(p.x-l.a.x) && !zero(p.x-l.b.x); }

int same(point p1,point p2,line l){ return xmult(p1,l.a,l.b)\*xmult(p2,l.a,l.b)>eps; }

int diff(point p1,point p2,line l){ return xmult(p1,l.a,l.b)\*xmult(p2,l.a,l.b)<-eps; }

int pingxing(line u,line v){ return zero(xmult((point){0,0},u.b-u.a,v.b-v.a)); }

int chuizhi(line u,line v){ return zero(dmult((point){0,0},u.b-u.a,v.b-v.a)); }

/\*[cos -sin]

[sin cos]

\*/

int jiao1(line u,line v){//判两线段相交,包括端点和部分重合

if (!dotinline(u.a,u.b,v.a)||!dotinline(u.a,u.b,v.b))

return !same(u.a,u.b,v)&&!same(v.a,v.b,u);

return dot\_online\_in(u.a,v)||dot\_online\_in(u.b,v)||dot\_online\_in(v.a,u)||dot\_online\_in(v.b,u);

}

int jiao2(line u,line v){ return diff(u.a,u.b,v)&&diff(v.a,v.b,u); }//判两线段相交,不包括端点和部分重合

double disdotline(point p,line l){ return abs(xmult(p,l.a,l.b))/dist(l.a,l.b); }//点到直线距离

double dis(point p,line l){//点到线段距离

point t={p.x+l.a.y-l.b.y,p.y+l.b.x-l.a.x};

if (xmult(l.a,t,p)\*xmult(l.b,t,p)>eps) return min(dist(p,l.a),dist(p,l.b));

return abs(xmult(p,l.a,l.b))/dist(l.a,l.b);

}

struct Line{ double a,b,c; };

point duicheng(point p1,point p2){ return (point){2\*p2.x-p1.x,2\*p2.y-p1.y}; } ////求p1关于p2的对称点

point duicheng2(point p,Line l){//p点关于直线L的对称点

double d=l.a\*l.a+l.b\*l.b;

return (point){ l.b\*l.b\*p.x-l.a\*l.a\*p.x-2\*l.a\*l.b\*p.y-2\*l.a\*l.c/d,

l.a\*l.a\*p.y-l.b\*l.b\*p.y-2\*l.a\*l.b\*p.x-2\*l.b\*l.c/d };

}

Line toline(point p1,point p2){ return (Line){ p2.y-p1.y,p1.x-p2.x,p2.x\*p1.y-p1.x\*p2.y}; }

//两点式化为一般式

point interline(Line l1,Line l2){//求直线的交点，注意平行的情况无解，避免RE

double a1=l1.a,b1=l1.b,c1=l1.c,a2=l2.a,b2=l2.b,c2=l2.c,x,y;

if (sign(b1)==0) x=-c1/a1,y=(-c2-a2\*x)/b2;

else x=(c1\*b2-b1\*c2)/(b1\*a2-b2\*a1),y=(-c1-a1\*x)/b1;

return (point){x,y};

}

P work(P p1,P p2,P p3,P p4){//两线段交点

double t1=cross(p1,p2,p3),t2=cross(p1,p2,p4);

if (sgn(t1)\*sgn(t2)<0){

double x=(fabs(t2)\*p3.x+fabs(t1)\*p4.x)/(fabs(t1)+fabs(t2));

double y=(fabs(t2)\*p3.y+fabs(t1)\*p4.y)/(fabs(t1)+fabs(t2));

return (P){x,y};

}

}

最小圆覆盖：

//O(n^4)

#include<cstdio>

#include<cmath>

#include<algorithm>

#define rep(i,l,r) for (i=l; i<=r; i++)

using namespace std;

const double eps=1e-8;

int i,j,k,l,f,n;

double r,ans=1e10;

struct P{ double x,y; }p[510],s,t;

double dist(P A,P B){ return sqrt((A.x-B.x)\*(A.x-B.x)+(A.y-B.y)\*(A.y-B.y)); }

P circenter(P A,P B,P C)

{

double a1=B.x-A.x,b1=B.y-A.y,c1=(a1\*a1+b1\*b1)/2;

double a2=C.x-A.x,b2=C.y-A.y,c2=(a2\*a2+b2\*b2)/2;

double d=a1\*b2-a2\*b1;

return (P){A.x+(c1\*b2-c2\*b1)/d,A.y+(a1\*c2-a2\*c1)/d};

}

int main(){

scanf("%d",&n);

rep(i,1,n) scanf("%lf%lf",&p[i].x,&p[i].y);

if (n==1) { printf("%.2lf %.2lf\n%.2lf",p[1].x,p[1].y,0.0); return 0; }

rep(i,1,n-1)

rep(j,i+1,n){

s=(P){(p[i].x+p[j].x)/2,(p[i].y+p[j].y)/2};

r=dist(s,p[i]); f=1;

rep(k,1,n) if (dist(p[k],s)-eps>r) { f=0; break; }

if (f && r<ans) ans=r,t=s;

}

rep(i,1,n-2)

rep(j,i+1,n-1)

rep(k,j+1,n)

if (fabs((p[j].x-p[i].x)\*(p[k].y-p[i].y)-(p[k].x-p[i].x)\*(p[j].y-p[i].y))>eps){

s=circenter(p[i],p[j],p[k]);

r=dist(s,p[i]); f=1;

rep(l,1,n) if (dist(s,p[l])-eps>r) { f=0; break; }

if (f && r<ans) ans=r,t=s;

}

printf("%.2lf %.2lf %.2lf",t.x,t.y,ans);

return 0;

}

//O(n)

#include<cstdio>

#include<cmath>

#include<algorithm>

#define rep(i,l,r) for (i=l; i<=r; i++)

using namespace std;

const double eps=1e-8;

struct P{ double x,y; }p[1010],c,d;

int i,j,k,n;

double r;

double dist(P A,P B){ return sqrt((A.x-B.x)\*(A.x-B.x)+(A.y-B.y)\*(A.y-B.y)); }

P circenter(P A,P B,P C){

double a1=B.x-A.x,b1=B.y-A.y,c1=(a1\*a1+b1\*b1)/2;

double a2=C.x-A.x,b2=C.y-A.y,c2=(a2\*a2+b2\*b2)/2;

double d=a1\*b2-a2\*b1;

return (P){A.x+(c1\*b2-c2\*b1)/d,A.y+(a1\*c2-a2\*c1)/d};

}

void work(P \*p,int n,P &c,double &r){

random\_shuffle(p+1,p+n+1); c=p[1]; r=0;

rep(i,2,n)

if (dist(p[i],c)>r+eps){

c=p[i]; r=0;

rep(j,1,i-1)

if (dist(p[j],c)>r+eps){

c=(P){(p[i].x+p[j].x)/2,(p[i].y+p[j].y)/2};

r=dist(p[j],c);

rep(k,1,j-1)

if (dist(p[k],c)>r+eps) c=circenter(p[i],p[j],p[k]),r=dist(p[i],c);

}

}

}

int main(){

for ( scanf("%d",&n); n; scanf("%d",&n)){

rep(i,1,n) scanf("%lf%lf",&p[i].x,&p[i].y);

work(p,n,c,r); printf("%.2f %.2f %.2f\n",c.x,c.y,r);

}

return 0;

}

半平面交：

//O(n^2)

#include<cstdio>

#include<cmath>

#include<algorithm>

#define rep(i,l,r) for (i=l; i<=r; i++)

using namespace std;

const int N=210,inf=0x3f3f3f3f;

const double eps=1e-8;

inline double sgn(double x){ return (fabs(x)<eps) ? 0 : ( x>0 ? 1 : -1); }

int i,T,ntot,n,tot;

struct P{ double x,y; }p[N],a[N],s[N];

struct S{ P s,t; };

double cross(P a,P b,P c) { return (b.x-a.x)\*(c.y-a.y)-(b.y-a.y)\*(c.x-a.x); }

bool outside(S seg,P p){ return cross(seg.s,seg.t,p)>eps; }

bool inside(S seg,P p){ return cross(seg.s,seg.t,p)<-eps; }

void work(P p1,P p2,P p3,P p4){

double t1=cross(p1,p2,p3),t2=cross(p1,p2,p4);

if (sgn(t1)\*sgn(t2)<0){

double x=(fabs(t2)\*p3.x+fabs(t1)\*p4.x)/(fabs(t1)+fabs(t2));

double y=(fabs(t2)\*p3.y+fabs(t1)\*p4.y)/(fabs(t1)+fabs(t2));

a[++tot]=(P){x,y};

}

}

void cut(S seg){

int i; tot=0;

rep(i,1,ntot){

if (!outside(seg,p[i])) a[++tot]=p[i];

else work(seg.s,seg.t,p[i-1],p[i]),work(seg.s,seg.t,p[i],p[i+1]);

}

ntot=tot; swap(a,p); p[0]=p[tot]; p[tot+1]=p[1];

}

void solve(){

rep(i,1,n) scanf("%lf%lf",&s[i].x,&s[i].y); s[n+1]=s[1];

p[1]=(P){-inf,-inf}; p[2]=(P){-inf,inf}; p[3]=(P){inf,inf}; p[4]=(P){inf,-inf};

ntot=4; p[0]=p[4]; p[5]=p[1];

rep(i,1,n) cut((S){s[i],s[i+1]});

if (tot==0) puts("NO"); else puts("YES");

}

//O(nlogn)

#include<cmath>

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=100010;

const double eps=1e-8;

struct P{

double x,y;

P(double xx=0,double yy=0):x(xx),y(yy){}

}p[N],a[N];

struct L{P a,b; double slop;}l[N],q[N];

int n,k,cnt,tot;

double ans1,ans2;

P operator -(P a,P b){ return P(a.x-b.x,a.y-b.y); }

double operator \*(P a,P b){ return a.x\*b.y-a.y\*b.x; }

bool operator <(L a,L b){ return (a.slop!=b.slop) ? (a.slop-b.slop)<-eps : (a.b-a.a)\*(b.b-a.a)>eps; }

P inter(L a,L b){

double A1=a.b.y-a.a.y,B1=a.a.x-a.b.x,C1=(a.b.x-a.a.x)\*a.a.y-(a.b.y-a.a.y)\*a.a.x;

double A2=b.b.y-b.a.y,B2=b.a.x-b.b.x,C2=(b.b.x-b.a.x)\*b.a.y-(b.b.y-b.a.y)\*b.a.x;

return P((C2\*B1-C1\*B2)/(A1\*B2-A2\*B1),(C1\*A2-C2\*A1)/(A1\*B2-A2\*B1));

}

bool jud(L a,L b,L t){ P p=inter(a,b); return (t.b-t.a)\*(p-t.a)<-eps; }

void work(){

sort(l+1,l+cnt+1);

int L=1,R=0; tot=0;

rep(i,1,cnt){

if (abs(l[i].slop-l[i-1].slop)>eps) tot++;

l[tot]=l[i];

}

cnt=tot; tot=0;

q[++R]=l[1]; q[++R]=l[2];

rep(i,3,cnt){

while (L<R && jud(q[R-1],q[R],l[i])) R--;

while (L<R && jud(q[L+1],q[L],l[i])) L++;

q[++R]=l[i];

}

while (L<R && jud(q[R-1],q[R],q[L])) R--;

while (L<R && jud(q[L+1],q[L],q[R])) L++;

q[R+1]=q[L];

rep(i,L,R) a[++tot]=inter(q[i],q[i+1]);

}

void get1(){

ans1=0;

if (tot<3) return;

a[++tot]=a[1];

rep(i,1,tot) ans1+=a[i]\*a[i+1];

ans1=fabs(ans1)/2;

}

void get2(){

ans2=0; p[n+1]=p[1];

rep(i,1,n+1) ans2+=p[i]\*p[i+1];

ans2=fabs(ans2)/2;

}

void ins(int i,int j){

double a=p[1].y-p[i].y-p[2].y+p[j].y;

double b=p[2].x-p[j].x-p[1].x+p[i].x;

double c=p[1].x\*p[2].y-p[2].x\*p[1].y-p[i].x\*p[j].y+p[j].x\*p[i].y;

if (abs(a)<eps && abs(b)<eps) return;

if (abs(a)<eps){

if (b>0) l[++cnt].a=P(1,-c/b),l[cnt].b=P(0,-c/b);

else l[++cnt].a=P(0,-c/b),l[cnt].b=P(1,-c/b);

}

if (abs(b)<eps){

if (a>0) l[++cnt].a=P(-c/a,0),l[cnt].b=P(-c/a,1);

else l[++cnt].a=P(-c/a,1),l[cnt].b=P(-c/a,0);

}

if (abs(a)>eps && abs(b)>eps){

if (b>0) l[++cnt].a=P(1,-a/b-c/b),l[cnt].b=P(0,-c/b);

else l[++cnt].a=P(0,-c/b),l[cnt].b=P(1,-a/b-c/b);

}

}

int main(){

scanf("%d",&n);

rep(i,1,n) scanf("%lf%lf",&p[i].x,&p[i].y);

rep(i,2,n) ins(i,i%n+1); l[++cnt].a=p[1],l[cnt].b=p[2];

rep(i,1,cnt) l[i].slop=atan2(l[i].b.y-l[i].a.y,l[i].b.x-l[i].a.x);

work(); get1(); get2(); printf("%.4lf\n",ans1/ans2);

return 0;

}

旋转卡壳：

#include<cstdio>

#include<cmath>

#include<algorithm>

#define rep(i,l,r) for (i=l; i<=r; i++)

using namespace std;

const double eps=1e-5;

int i,n,top;

double ans,t;

struct P{ double x,y; }p[500010],q[500010];

double dis2(P a,P b){ return (a.x-b.x)\*(a.x-b.x)+(a.y-b.y)\*(a.y-b.y); }

double xmult(P a,P b,P c){ return (b.x-a.x)\*(c.y-a.y)-(b.y-a.y)\*(c.x-a.x);}

int cmp(P a,P b)

{ return (t=xmult(p[1],a,b))>eps || (t>-eps && dis2(a,p[1])<dis2(b,p[1])); }

void graham(){

int i,mx=1;

rep(i,2,n) if (p[i].y<p[mx].y || (p[i].y==p[mx].y && p[i].x<p[mx].x)) mx=i;

swap(p[1],p[mx]);

sort(p+2,p+n+1,cmp);

q[++top]=p[1]; q[++top]=p[2];

rep(i,3,n){

while (top>1 && xmult(q[top-1],q[top],p[i])<eps) top--;

q[++top]=p[i];

}

q[top+1]=q[1];

}

void RC(){

int i,x=2; ans=0;

rep(i,1,top){

while (xmult(q[i],q[i+1],q[x])<xmult(q[i],q[i+1],q[x+1])) x=(x==top) ? 1 : x+1;

ans=max(ans,dis2(q[x],q[i]));

}

}

int main(){

scanf("%d",&n);

rep(i,1,n) scanf("%lf%lf",&p[i].x,&p[i].y);

graham(); RC(); printf("%d",(int)ans);

return 0;

}

点分治：

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (i=l; i<=r; i++)

#define For(i,x) for (i=h[x]; i; i=e[i].nxt) if (!vis[k=e[i].to] && k!=fa)

using namespace std;

const int N=20010,inf=0x3f3f3f3f;

int i,n,x,y,v,cnt,k,tot,rt,sum,ans,fa=0,h[N],f[N],vis[N],sz[N],d[N],a[N];

struct E{ int to,val,nxt; } e[N];

void add(int u,int v,int w){ e[++cnt]=(E){v,w,h[u]}; h[u]=cnt; }

void get(int x,int fa){

int i,k; sz[x]=1; f[x]=0;

For(i,x) get(k,x),sz[x]+=sz[k],f[x]=max(f[x],sz[k]);

f[x]=max(f[x],sum-sz[x]);

if (f[x]<f[rt]) rt=x;

}

void deep(int x,int fa){

int i,k; a[++tot]=d[x];

For(i,x) d[k]=d[x]+e[i].val,deep(k,x);

}

int cal(int x,int v){

d[x]=v; tot=0; deep(x,0);

sort(a+1,a+tot+1);

int l=1,r=tot,sum=0;

while (l<r)

if (a[l]+a[r]<=k) sum+=r-l,l++; else r--;

return sum;

}

void solve(int x){

int i,k; ans+=cal(x,0); vis[x]=1;

For(i,x){

ans-=cal(k,e[i].val); sum=sz[k];

rt=0; get(k,0); solve(rt);

}

}

int main(){

for (scanf("%d%d",&n,&k); n+k; scanf("%d%d",&n,&k)){

ans=rt=cnt=0; f[0]=inf; sum=n;

memset(vis,0,sizeof(vis)); memset(h,0,sizeof(h));

rep(i,1,n-1) scanf("%d%d%d",&x,&y,&v),add(x,y,v),add(y,x,v);

get(1,0); solve(rt); printf("%d\n",ans);

}

return 0;

}

可持久化Trie：

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=600100,M=300000\*2\*25;

char ch;

int n,m,nd,x,l,r,b[N],a[N],rt[N],c[M][2],sm[M];

void ins(int y,int &x,int k){

int tmp=++nd; x=nd;

for (int i=23; ~i; i--){

c[x][0]=c[y][0]; c[x][1]=c[y][1]; sm[x]=sm[y]+1;

int t=(k>>i)&1; c[x][t]=++nd; x=c[x][t]; y=c[y][t];

}

sm[x]=sm[y]+1; x=tmp;

}

int que(int l,int r,int k){

int res=0;

for (int i=23; ~i; i--){

int t=(k>>i)&1;

if (sm[c[r][t^1]]-sm[c[l][t^1]]) res+=1<<i,r=c[r][t^1],l=c[l][t^1];

else r=c[r][t],l=c[l][t];

}

return res;

}

int main(){

scanf("%d%d",&n,&m); n++; ins(rt[0],rt[1],0);

rep(i,2,n) scanf("%d",&a[i]),b[i]=b[i-1]^a[i],ins(rt[i-1],rt[i],b[i]);

rep(i,1,m){

scanf(" %c",&ch);

if (ch=='A') scanf("%d",&x),n++,b[n]=b[n-1]^x,ins(rt[n-1],rt[n],b[n]);

else scanf("%d%d%d",&l,&r,&x),printf("%d\n",que(rt[l-1],rt[r],b[n]^x));

}

return 0;

}

Splay：

#include<cstdio>

#include<algorithm>

#include<iostream>

#define rep(i,l,r) for (i=l; i<=r; i++)

#define L c[x][0]

#define R c[x][1]

#define key c[c[rt][1]][0]

using namespace std;

const int N=2000100,inf=1000000000;

int cnt,rt,x,n,pos=1,c[N][2],s[N],rev[N],f[N];

char v[N],ch[N],op[20];

void get(int &x,char ch,int fa){ x=++cnt; v[x]=ch; f[x]=fa; L=R=rev[x]=0; s[x]=1; }

void upd(int x){ if (x) s[x]=s[L]+s[R]+1; }

void make(int x){ if (x) rev[x]^=1,swap(L,R); }

void push(int x){ if (x && rev[x]) make(L),make(R),rev[x]=0; }

void rot(int &rt,int x){

int y=f[x],z=f[y],p=(c[y][1]==x),q=p^1;

if (y==rt) rt=x; else c[z][c[z][1]==y]=x;

f[x]=z; f[y]=x; f[c[x][q]]=y; c[y][p]=c[x][q]; c[x][q]=y;

upd(y);

}

void splay(int &rt,int x){

while (x!=rt){

int y=f[x],z=f[y];

if (y!=rt)

{ if ((c[z][0]==y) ^ (c[y][0]==x)) rot(rt,x); else rot(rt,y); }

rot(rt,x);

}

upd(x);

}

int find(int x,int k){

push(x);

if (s[L]+1==k) return x;

else return (s[L]>=k) ? find(L,k) : find(R,k-s[L]-1);

}

inline void split(int l,int r){ int x=find(rt,l-1),y=find(rt,r+1); splay(rt,x); splay(c[rt][1],y); }

void insert(int &k,int l,int r,int fa){

int mid=(l+r)>>1; get(k,ch[mid],fa);

if (l<mid) insert(c[k][0],l,mid-1,k);

if (mid<r) insert(c[k][1],mid+1,r,k);

upd(k);

}

void ins(){

scanf("%d%\*c",&x); gets(ch); split(pos+1,pos);

insert(key,0,x-1,c[rt][1]); upd(c[rt][1]); upd(rt);

}

void del(){ scanf("%d",&x); split(pos+1,pos+x); f[key]=0; key=0; upd(c[rt][1]); upd(rt); }

void reverse(){ scanf("%d",&x); split(pos+1,pos+x); make(key); }

int main(){

get(rt,'\*',0); get(c[rt][1],'\*',rt); upd(rt);

for (scanf("%d",&n); n--; ){

scanf("%s",op);

if (op[0]=='M') scanf("%d",&x),pos=x+1;

if (op[0]=='I') ins();

if (op[0]=='D') del();

if (op[0]=='R') reverse();

if (op[0]=='G') printf("%c\n",v[find(rt,pos+1)]);

if (op[0]=='P') pos--;

if (op[0]=='N') pos++;

}

return 0;

}

LCT:

#include<cstdio>

#include<cctype>

#include<algorithm>

#define ls ch[x][0]

#define rs ch[x][1]

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

inline void rd(int &x){

char ch; int t;

for (t=0; !isdigit(ch=getchar()); t=(ch=='-'));

for (x=ch-'0'; isdigit(ch=getchar()); x=x\*10+ch-'0');

if (t) x=-x;

}

const int N=50100,M=100100,inf=1000000000;

int n,m,x,y,a,b,t,fa[N+M],tmp[N+M],ans=inf;

struct edge{ int x,y,a,b; }e[M];

bool operator <(edge a,edge b){ return a.a<b.a; }

int find(int x){ return fa[x]==x ? x : fa[x]=find(fa[x]);}

struct Tree{

int val[M+N],mx[M+N],f[M+N],ch[M+N][2],rev[M+N];

int isroot(int x) { return (!f[x])||(ch[f[x]][0]!=x && ch[f[x]][1]!=x);}

void rev1(int x) { swap(ls,rs); rev[x]^=1;}

void push(int x) { if (rev[x]) rev1(ls),rev1(rs),rev[x]=0;}

void upd(int x){

mx[x]=x;

if (ls && val[mx[x]]<val[mx[ls]]) mx[x]=mx[ls];

if (rs && val[mx[x]]<val[mx[rs]]) mx[x]=mx[rs];

}

void rot(int x){

int y=f[x],z=f[y],w=ch[y][1]==x;

ch[y][w]=ch[x][w^1]; f[ch[x][w^1]]=y;

if (!isroot(y)) ch[z][ch[z][1]==y]=x;

f[x]=z; f[y]=x; ch[x][w^1]=y; upd(y);

}

void splay(int x){

int s=1,i=x; tmp[1]=x;

while (!isroot(i)) tmp[++s]=i=f[i];

while (s) push(tmp[s--]);

while (!isroot(x)){

int y=f[x];

if (!isroot(y))

((ch[f[y]][1]==y)^(ch[y][1]==x)) ? rot(x) :rot(y);

rot(x);

}

upd(x);

}

void access(int x){ for (int y=0; x; y=x,x=f[x]) splay(x),ch[x][1]=y,upd(x); }

void mkroot(int x){ access(x); splay(x); rev1(x);}

void link(int x,int y){ mkroot(x); f[x]=y; }

void cut(int x,int y){ mkroot(x); access(y); splay(y); ch[y][0]=f[x]=0; upd(y); }

int que(int x,int y){ mkroot(x); access(y); splay(y); return mx[y]; }

int find(int x){

access(x); splay(x);

while (ls) x=ls;

return x;

}

}T;

int main(){

rd(n); rd(m); rep(i,1,n) fa[i]=i;

rep(i,1,m) rd(e[i].x),rd(e[i].y),rd(e[i].a),rd(e[i].b);

sort(e+1,e+m+1);

rep(i,1,m){

x=e[i].x; y=e[i].y; a=e[i].a; b=e[i].b;

if (find(x)==find(y)){

t=T.que(x,y);

if (b<T.val[t]) T.cut(t,e[t-n].x),T.cut(t,e[t-n].y);

else continue;

}

else fa[find(x)]=find(y);

T.val[n+i]=b; T.mx[n+i]=n+i; T.link(x,n+i); T.link(y,n+i);

if (find(1)==find(n)) ans=min(ans,T.val[T.que(1,n)]+a);

}

if (ans==inf) puts("-1"); else printf("%d\n",ans);

return 0;

}

无旋treap：

#include<cstdio>

#include<algorithm>

#define P pair<int,int>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=200100,inf=1000000000;

int T,nd,rt,op,ans,x,ls[N],rs[N],h[N],sz[N],v[N];

void upd(int x){ sz[x]=sz[ls[x]]+sz[rs[x]]+1; }

int get(int x){ v[++nd]=x; sz[nd]=1; h[nd]=rand(); return nd; }

int merge(int x,int y){

if (!x || !y) return x+y;

if (h[x]<h[y]) { rs[x]=merge(rs[x],y); upd(x); return x; }

else { ls[y]=merge(x,ls[y]); upd(y); return y; }

}

P split(int x,int k){

if (!x) return P(0,0);

P tmp;

if (k<=sz[ls[x]]) tmp=split(ls[x],k),ls[x]=tmp.second,upd(x),tmp=P(tmp.first,x);

else tmp=split(rs[x],k-sz[ls[x]]-1),rs[x]=tmp.first,upd(x),tmp=P(x,tmp.second);

return tmp;

}

int rank(int x,int k){

if (!x) return 0;

if (v[x]>=k) return rank(ls[x],k); else return rank(rs[x],k)+sz[ls[x]]+1;

}

int find(int x,int k){

if (!x) return 0;

if (k==sz[ls[x]]+1) return x;

if (k<=sz[ls[x]]) return find(ls[x],k); else return find(rs[x],k-sz[ls[x]]-1);

}

void ins(int k){

int v=rank(rt,k); P x=split(rt,v);

rt=merge(merge(x.first,get(k)),x.second);

}

void del(int k){

int v=rank(rt,k);

P x=split(rt,v),y=split(x.second,1);

rt=merge(x.first,y.second);

}

int main(){

for (scanf("%d",&T); T--; ){

scanf("%d%d",&op,&x);

if (op==1) ins(x);

if (op==2) del(x);

if (op==3) printf("%d\n",rank(rt,x)+1);

if (op==4) printf("%d\n",v[find(rt,x)]);

if (op==5) printf("%d\n",v[find(rt,rank(rt,x))]);

if (op==6) printf("%d\n",v[find(rt,rank(rt,x+1)+1)]);

}

return 0;

}

虚树：

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long ll;

using namespace std;

const int N=1000010,inf=1e9;

int n,q,dis[N],dfn[N],tot,dep[N],fa[N][23],num,p[N];

int stk[N],top,siz[N],maxs[N],mins[N],ans1,ans2;

ll f[N]; bool bo[N];

bool cmp(int a,int b){return dfn[a]<dfn[b];}

struct Tgraph{

int nxt[N<<1],h[N],to[N<<1],cnt;

void add(int a,int b) {nxt[++cnt]=h[a]; h[a]=cnt; to[cnt]=b; }

void dfs1(int x){

dfn[x]=++tot;

for (int i=1;i<=20;i++) fa[x][i]=fa[fa[x][i-1]][i-1];

for (int i=h[x],k; i; i=nxt[i]) if ((k=to[i])!=fa[x][0]){

dis[k]=dis[x]+1; dep[k]=dep[x]+1; fa[k][0]=x; dfs1(k);

}

}

void dfs2(int x){

siz[x]=bo[x]; maxs[x]=0; mins[x]=inf; f[x]=0;

for (int i=h[x],k; i; i=nxt[i]){

int d=dis[k=to[i]]-dis[x];

dfs2(k),siz[x]+=siz[k];

ans1=min(ans1,mins[x]+mins[k]+d),mins[x]=min(mins[x],mins[k]+d);

ans2=max(ans2,maxs[x]+maxs[k]+d),maxs[x]=max(maxs[x],maxs[k]+d);

f[x]+=f[k]+1ll\*siz[k]\*(num-siz[k])\*d;

}

if (bo[x]) ans1=min(ans1,mins[x]),ans2=max(ans2,maxs[x]),mins[x]=0;

h[x]=0;

}

}g1,g2;

int lca(int a,int b){

if (dep[a]<dep[b]) swap(a,b);

for (int h=dep[a]-dep[b],i=20;i>=0;i--)

if (h>=(1<<i)) h-=(1<<i),a=fa[a][i];

for (int i=20;i>=0;i--) if (fa[a][i]!=fa[b][i]) a=fa[a][i],b=fa[b][i];

if (a==b) return a; else return fa[a][0];

}

void work(){

top=0;

rep(i,1,num){

if (!top) { stk[++top]=p[i]; continue; }

int u=lca(stk[top],p[i]);

while (dfn[u]<dfn[stk[top]]){

if (dfn[u]>=dfn[stk[top-1]]){

g2.add(u,stk[top]);

if (stk[--top]!=u) stk[++top]=u;

break;

}

g2.add(stk[top-1],stk[top]); top--;

}

stk[++top]=p[i];

}

while (top>1) g2.add(stk[top-1],stk[top]),top--;

ans1=inf; ans2=0; g2.dfs2(stk[1]);

printf("%lld %d %d\n",f[stk[1]],ans1,ans2);

rep(i,1,num) bo[p[i]]=0;g2.cnt=0;

}

int main(){

scanf("%d",&n);

for (int i=1,a,b;i<n;i++) scanf("%d%d",&a,&b),g1.add(a,b),g1.add(b,a);

g1.dfs1(1); scanf("%d",&q);

for (int i=1;i<=q;i++){

scanf("%d",&num);

for (int j=1;j<=num;j++) scanf("%d",&p[j]),bo[p[j]]=1;

sort(p+1,p+1+num,cmp); work();

}

return 0;

}

左偏树：

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=100100;

struct T{ int l,r,v,dis,f; }h[N];

int a,b,A,B,n,m;

int merge(int a,int b){

if (!a||!b) return a+b;

if (h[a].v<h[b].v) swap(a,b);

h[a].r=merge(h[a].r,b); h[h[a].r].f=a;

if (h[h[a].l].dis<h[h[a].r].dis) swap(h[a].l,h[a].r);

h[a].dis=(h[a].r==0) ? 0 : h[h[a].r].dis+1;

return a;

}

int pop(int a){

int l=h[a].l,r=h[a].r;

h[l].f=l; h[r].f=r; h[a].l=h[a].r=h[a].dis=0;

return merge(l,r);

}

int find(int a){ return h[a].f==a ? a : find(h[a].f); }

int main(){

while (scanf("%d",&n)==1){

rep(i,1,n) scanf("%d",&h[i].v),h[i].l=h[i].r=h[i].dis=0,h[i].f=i;

for (scanf("%d",&m); m--; ){

scanf("%d%d",&a,&b); A=find(a); B=find(b);

if (A==B) puts("-1");

else{

h[A].v/=2; int u=pop(A); u=merge(u,A);

h[B].v/=2; int v=pop(B); v=merge(v,B);

printf("%d\n",h[merge(u,v)].v);

}

}

}

return 0;

}

块状：

#include<cstdio>

#include<cstring>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long LL;

using namespace std;

const int N=1<<25,S=20000,M=N/S\*3;

int n,m,cur,cnt=1,stk[M+110];

char str[N+110],s[100];

struct node{ char data[S+110]; int len,nxt; }a[M+110];

int get(){ return stk[cnt++]; }

void dis(int x){ stk[--cnt]=x; }

void init(){ a[0].len=0; a[0].nxt=-1; }

void find(int &pos,int &now)

{ for (now=0; a[now].nxt!=-1 && pos>a[now].len; now=a[now].nxt) pos-=a[now].len;}

void fill(int pos,int n,char\* data,int nxt)

{ a[pos].nxt=nxt; a[pos].len=n; memcpy(a[pos].data,data,n);}

void split(int pos,int p){

if (a[pos].len==p) return;

int t=get();

fill(t,a[pos].len-p,a[pos].data+p,a[pos].nxt);

a[pos].nxt=t; a[pos].len=p;

}

void maintain(int pos){

for (int t; pos!=-1; pos=a[pos].nxt)

for (t=a[pos].nxt; t!=-1 && a[pos].len+a[t].len<S; t=a[pos].nxt){

memcpy(a[pos].data+a[pos].len,a[t].data,a[t].len);

a[pos].len+=a[t].len; a[pos].nxt=a[t].nxt; dis(t);

}

}

void ins(int pos,int n){

int now,i,t;

find(pos,now); split(now,pos);

for (i=0; i+S<=n; i+=S,now=t)

{ t=get(); fill(t,S,str+i,a[now].nxt); a[now].nxt=t;}

if (i<n)

{ t=get(); fill(t,n-i,str+i,a[now].nxt); a[now].nxt=t;}

maintain(now);

}

void del(int pos,int n){

int now,i,t;

find(pos,now); split(now,pos);

for (i=a[now].nxt; i!=-1 && a[i].len<n; i=a[i].nxt) n-=a[i].len;

split(i,n); i=a[i].nxt;

for (t=a[now].nxt; t!=i; t=a[t].nxt) dis(t);

a[now].nxt=i; maintain(now);

}

void put(int pos,int n){

int now,i,t;

find(pos,now); i=min(n,a[now].len-pos); memcpy(str,a[now].data+pos,i);

for (t=a[now].nxt; t!=-1 && i+a[t].len<=n; i+=a[t].len,t=a[t].nxt)

memcpy(str+i,a[t].data,a[t].len);

if (i<n && t!=-1) memcpy(str+i,a[t].data,n-i);

str[n]=0;

}

void rd(int m){

for (int i=-1; i<m-1;){

char c=getchar(); str[++i]=c;

if (c<32||c>126) i--;

}

}

int main(){

init(); scanf("%d",&n);

rep(i,1,M+100) stk[i]=i;

rep(i,1,n){

scanf("%s",s);

if (s[0]=='M') scanf("%d",&cur);

if (s[0]=='I') scanf("%d",&m),rd(m),ins(cur,m);

if (s[0]=='N') cur++;

if (s[0]=='P') cur--;

if (s[0]=='D') scanf("%d",&m),del(cur,m);

if (s[0]=='G') scanf("%d",&m),put(cur,m),puts(str);

}

}

//块状树:

#include<cstdio>

#include<cctype>

#include<cmath>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

typedef long long LL;

using namespace std;

const int inf=1000000000,N=30100,M=60100;

int n,q,u,v,lim,own[N],dep[N],fa[N],mx[N],sz[N],sum[N],w[N];

struct E{ int v,nxt; };

struct Edge{

int cnt,h[N]; E e[M];

void add(int x,int y) { e[++cnt]=(E){y,h[x]}; h[x]=cnt;}

}G,G1;

template<typename T>void rd(T &x){

char ch; int t;

for (t=0; !isdigit(ch=getchar()); t=(ch=='-'));

for (x=ch-'0'; isdigit(ch=getchar()); x=x\*10+ch-'0');

if (t) x=-x;

}

void build(int u,int f,int d){

dep[u]=d; fa[u]=f; int v;

if (!own[u]) sz[own[u]=u]++;

for (int i=G.h[u]; i; i=G.e[i].nxt)

if ((v=G.e[i].v)!=f){

if (sz[own[u]]<lim) G1.add(u,v),sz[own[v]=own[u]]++;

build(v,u,d+1);

}

}

void dfs(int u,int s,int m){

sum[u]=s+=w[u]; mx[u]=m=max(m,w[u]);

for (int i=G1.h[u]; i; i=G1.e[i].nxt) dfs(G1.e[i].v,s,m);

}

void change(int u,int k){

w[u]=k;

if (own[u]==u) dfs(u,0,-inf); else dfs(u,sum[fa[u]],mx[fa[u]]);

}

pair<int,int> que(int u,int v){

int sm=0,m=-inf;

while (u!=v){

if (dep[u]<dep[v]) swap(u,v);

if (own[u]==own[v]) sm+=w[u],m=max(m,w[u]),u=fa[u];

else{

if (dep[own[u]]<dep[own[v]]) swap(u,v);

sm+=sum[u],m=max(m,mx[u]),u=fa[own[u]];

}

}

return make\_pair(sm+w[u],max(m,w[u]));

}

int main(){

rd(n); lim=sqrt(n)+1;

rep(i,1,n-1) rd(u),rd(v),G.add(u,v),G.add(v,u);

rep(i,1,n) rd(w[i]); build(1,0,1);

rep(i,1,n) if (own[i]==i) dfs(i,0,-inf);

rd(q); char opt[10];

while (q--){

scanf("%s\n",opt); rd(u); rd(v);

if (opt[0]=='C') change(u,v);

else if (opt[1]=='M') printf("%d\n",que(u,v).second);

else printf("%d\n",que(u,v).first);

}

return 0;

}

树剖：

#include<cstdio>

#include<algorithm>

#define lc x<<1

#define rc ((x<<1)|1)

#define rep(i,l,r) for (int i=l; i<=r; i++)

#define For(i,x) for (i=h[x]; i; i=e[i].nxt)

using namespace std;

const int N=100100,inf=2147483647;

int n,m,u,v,k,op,cnt,x,y,z,rt,d[N],son[N],sz[N],f[N];

Int h[N],last[N],top[N],w[N],cov[N<<2],mn[N<<2];

struct E{ int to,nxt; }e[N<<1];

void add(int u,int v){ e[++cnt]=(E){v,h[u]}; h[u]=cnt; }

void dfs(int x){

int i,k; sz[x]=1;

For(i,x) if ((k=e[i].to)!=f[x]){

d[k]=d[x]+1; f[k]=x; dfs(k); sz[x]+=sz[k];

if (sz[k]>sz[son[x]]) son[x]=k;

}

}

void dfs2(int x,int t){

int i,k; top[x]=t; w[x]=++z;

if (son[x]) dfs2(son[x],t);

For(i,x) if ((k=e[i].to)!=f[x] && k!=son[x]) dfs2(k,k);

last[x]=z;

}

int get(int x,int y){

while (f[top[y]]!=x && top[x]!=top[y]) y=f[top[y]];

if (f[top[y]]!=x) return son[x]; else return top[y];

}

/\*

int lca(int x,int y)

for (; top[x]!=top[y]; x=f[top[x]]) if (d[top[x]]<d[top[y]]) swap(x,y);

return d[x]<d[y]?x:y;

}

\*/

void push(int x){

if (cov[x]) cov[lc]=cov[rc]=mn[lc]=mn[rc]=cov[x],cov[x]=0;

}

void mdf(int x,int l,int r,int L,int R,int k){

if (l==L && r==R) { mn[x]=cov[x]=k; return; }

int mid=(l+r)>>1; push(x);

if (R<=mid) mdf(lc,l,mid,L,R,k);

else if (L>mid) mdf(rc,mid+1,r,L,R,k);

else mdf(lc,l,mid,L,mid,k),mdf(rc,mid+1,r,mid+1,R,k);

mn[x]=min(mn[lc],mn[rc]);

}

int ask(int x,int l,int r,int L,int R){

if (l==L && r==R) return mn[x];

int mid=(l+r)>>1; push(x);

if (R<=mid) return ask(lc,l,mid,L,R);

if (L>mid) return ask(rc,mid+1,r,L,R);

return min(ask(lc,l,mid,L,mid),ask(rc,mid+1,r,mid+1,R));

}

void work(int x,int y,int z){

for ( ; top[x]!=top[y]; x=f[top[x]]){

if (d[top[x]]<d[top[y]]) swap(x,y);

mdf(1,1,n,w[top[x]],w[x],z);

}

if (d[x]<d[y]) swap(x,y);

mdf(1,1,n,w[y],w[x],z);

}

int main(){

scanf("%d%d",&n,&m);

rep(i,1,n-1) scanf("%d%d",&u,&v),add(u,v),add(v,u);

dfs(1); dfs2(1,1);

rep(i,1,n) scanf("%d",&k),mdf(1,1,n,w[i],w[i],k);

scanf("%d",&rt);

while (m--){

scanf("%d",&op);

if (op==1) scanf("%d",&rt);

if (op==2) scanf("%d%d%d",&x,&y,&z),work(x,y,z);

if (op==3){

scanf("%d",&x);

if (x==rt) printf("%d\n",ask(1,1,n,1,n));

else

if (f[rt]==x)

printf("%d\n",min(ask(1,1,n,1,w[rt]-1),(last[rt]==n) ? inf :ask(1,1,n,last[rt]+1,n)));

else if (w[rt]>=w[x] && w[rt]<=last[x])

y=get(x,rt),printf("%d\n",min(ask(1,1,n,1,w[y]-1),(last[y]==n) ? inf :ask(1,1,n,last[y]+1,n)));

else printf("%d\n",ask(1,1,n,w[x],last[x]));

}

}

return 0;

}

/\*

int work1(int x,int k){

if (v[x]>=k) return ll[x];

while (x)

if ((x&1)==0) x=x>>1;

else{ x=x>>1; if (mx[ls]>=k) break; }

if (!x) return 0;

for (x=ls; rr[x]-ll[x]; ) if (mx[rs]>=k) x=rs; else x=ls;

return ll[x];

}

int work2(int x,int k){

if (v[x]>=k) return ll[x];

while (x)

if (x&1) x=x>>1;

else{ x=x>>1; if (mx[rs]>=k) break; }

if (!x) return 0;

for (x=rs; rr[x]-ll[x]; ) if (mx[ls]>=k) x=ls; else x=rs;

return ll[x];

}

\*/

KD-Tree：

#include<cstdio>

#include<cstdlib>

#include<algorithm>

#define lson ls[x],L,mid

#define rson rs[x],mid+1,R

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const double A=0.75;

const int N=5000000,M=1000000000;

int ans,D,lst,n,q,x,y,k,top,tot,need,dd,nd,nd1,X1,Y1,X2,Y2,op,Rt,ls[N],rs[N],rt[N],stk[N],w[N];

struct P{

int d[2],mn[2],mx[2],l,r,sz;

int& operator [](int x){ return d[x]; }

friend bool operator <(P a,P b){ return a[D]<b[D]; }

void clear(){ d[0]=d[1]=mn[0]=mn[1]=mx[0]=mx[1]=l=r=sz=0; }

}t[N],p[N],T;

int get(){ if (top) return x=stk[top--]; else return ++nd; }

void del(int x){ t[x].clear(); stk[++top]=x; }

void upd(int x){

int l=t[x].l,r=t[x].r; t[x].sz=1;

if (l) t[x].sz+=t[l].sz;

if (r) t[x].sz+=t[r].sz;

rep(i,0,1){

t[x].mn[i]=t[x].mx[i]=t[x][i];

if (l) t[x].mn[i]=min(t[x].mn[i],t[l].mn[i]),t[x].mx[i]=max(t[x].mx[i],t[l].mx[i]);

if (r) t[x].mn[i]=min(t[x].mn[i],t[r].mn[i]),t[x].mx[i]=max(t[x].mx[i],t[r].mx[i]);

}

}

int build(int l,int r,int k){

if (l>r) return 0;

D=k; int mid=(l+r)>>1,x=get();

nth\_element(p+l,p+mid,p+r+1); t[x]=p[mid];

t[x].l=build(l,mid-1,k^1); t[x].r=build(mid+1,r,k^1);

upd(x); return x;

}

bool in(int x1,int y1,int x2,int y2,int X1,int Y1,int X2,int Y2)

{ return x1<=X1 && X2<=x2 && y1<=Y1 && Y2<=y2;}

bool out(int x1,int y1,int x2,int y2,int X1,int Y1,int X2,int Y2)

{ return x1>X2 || x2<X1 || y1>Y2 || y2<Y1; }

int dfs(int x){ if (t[x].l) dfs(t[x].l); p[++tot]=t[x]; if (t[x].r) dfs(t[x].r); del(x);}

void ins(int &x,int k){

if (!x) { x=get(); t[x]=T; upd(x); return; }

if (T[k]<t[x][k]) ins(t[x].l,k^1); else ins(t[x].r,k^1);

if (t[t[x].l].sz>t[x].sz\*A || t[t[x].r].sz>t[x].sz\*A) need=x,dd=k;

upd(x);

}

void ins1(int &rt){

need=0; ins(rt,0);

if (need) tot=0,dfs(need),need=build(1,tot,dd);

}

void que(int x,int k){

if (!x || out(X1,Y1,X2,Y2,t[x].mn[0],t[x].mn[1],t[x].mx[0],t[x].mx[1])) return;

if (in(X1,Y1,X2,Y2,t[x].mn[0],t[x].mn[1],t[x].mx[0],t[x].mx[1])) { ans+=t[x].sz; return; }

if (in(X1,Y1,X2,Y2,t[x][0],t[x][1],t[x][0],t[x][1])) ans++;

que(t[x].l,k^1); que(t[x].r,k^1);

}

void insert(int &x,int L,int R,int v){

if (!x) x=++nd1;

ins1(rt[x]); int mid=(L+R)>>1;

if (L==R) return;

if (v<=mid) insert(lson,v); else insert(rson,v);

}

int query(int x,int L,int R,int k){

if (L==R) return L;

ans=0; if (rs[x]) que(rt[rs[x]],0); int mid=(L+R)>>1;

if (k>ans) return query(lson,k-ans); else return query(rson,k);

}

int main(){

scanf("%d%d",&n,&q);

rep(i,1,N-1) w[i]=10000;

rep(i,1,q){

scanf("%d",&op);

if (op==1)

scanf("%d%d%d",&x,&y,&k),x^=lst,y^=lst,k^=lst,T[0]=x,T[1]=y,insert(Rt,1,M,k);

else{

scanf("%d%d%d%d%d",&X1,&Y1,&X2,&Y2,&k);

X1^=lst; Y1^=lst; X2^=lst; Y2^=lst; k^=lst;

ans=0; que(rt[Rt],0);

if (ans<k) { puts("NAIVE!ORZzyz."); lst=0; continue; }

printf("%d\n",lst=query(Rt,1,M,k));

}

}

return 0;

}

树套树：

#include<cstdio>

#include<cstdlib>

#include<algorithm>

#define mid ((L+R)>>1)

#define lc x<<1,L,mid

#define rc (x<<1)|1,mid+1,R

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=3000100,inf=100000000;

int n,m,x,y,k,tmp,sz,ans,rt[N],s[N],ls[N],rs[N],v[N],w[N],h[N],a[N];

void upd(int x){ s[x]=s[ls[x]]+s[rs[x]]+w[x]; }

void rrot(int &x){ int y=ls[x]; ls[x]=rs[y]; rs[y]=x; s[y]=s[x]; upd(x); x=y; }

void lrot(int &x){ int y=rs[x]; rs[x]=ls[y]; ls[y]=x; s[y]=s[x]; upd(x); x=y; }

void ins(int &x,int k){

if (!x) { x=++sz; s[x]=w[x]=1; v[x]=k; h[x]=rand(); return; }

s[x]++;

if (k==v[x]) w[x]++;

else if (k<v[x]) { ins(ls[x],k); if (h[ls[x]]<h[x]) rrot(x); }

else { ins(rs[x],k); if (h[rs[x]]<h[x]) lrot(x); }

}

void del(int &x,int k){

if (v[x]==k){

if (w[x]>1) { w[x]--; s[x]--; return; }

if ((!ls[x]) || (!rs[x])) { x=ls[x]+rs[x]; return; }

if (h[ls[x]]<h[rs[x]]) { rrot(x); del(x,k); } else { lrot(x); del(x,k); }

return;

}

if (k<v[x]) { del(ls[x],k); s[x]--; } else { del(rs[x],k); s[x]--; }

}

void askrank(int x,int k){

if (!x) return;

if (k==v[x]) { tmp+=s[ls[x]]; return; }

else if (k<v[x]) askrank(ls[x],k); else tmp+=s[ls[x]]+w[x],askrank(rs[x],k);

}

void pre(int x,int k){

if (!x) return;

if (v[x]<k) tmp=max(v[x],tmp),pre(rs[x],k); else pre(ls[x],k);

}

void nxt(int x,int k){

if (!x) return;

if (v[x]>k) tmp=min(v[x],tmp),nxt(ls[x],k); else nxt(rs[x],k);

}

void build(int x,int L,int R,int pos,int k){

ins(rt[x],k);

if (L==R) return;

if (pos<=mid) build(lc,pos,k); else build(rc,pos,k);

}

void getrank(int x,int L,int R,int l,int r,int k){

if (L==l && R==r) { askrank(rt[x],k); return; }

if (r<=mid) getrank(lc,l,r,k);

else if (l>mid) getrank(rc,l,r,k);

else getrank(lc,l,mid,k),getrank(rc,mid+1,r,k);

}

void getidx(int x,int y,int z){

int l=0,r=inf,ans;

while (l<=r){

int m=(l+r)>>1;

tmp=1; getrank(1,1,n,x,y,m);

if (tmp<=z) l=m+1,ans=m; else r=m-1;

}

printf("%d\n",ans);

}

void change(int x,int L,int R,int pos,int k,int y){

del(rt[x],y); ins(rt[x],k);

if (L==R) return;

if (pos<=mid) change(lc,pos,k,y); else change(rc,pos,k,y);

}

void askpre(int x,int L,int R,int l,int r,int k){

if (L==l && R==r) { pre(rt[x],k); return; }

if (r<=mid) askpre(lc,l,r,k);

else if (l>mid) askpre(rc,l,r,k);

else askpre(lc,l,mid,k),askpre(rc,mid+1,r,k);

}

void asknxt(int x,int L,int R,int l,int r,int k){

if (L==l && R==r) { nxt(rt[x],k); return; }

if (r<=mid) asknxt(lc,l,r,k);

else if (l>mid) asknxt(rc,l,r,k);

else asknxt(lc,l,mid,k),asknxt(rc,mid+1,r,k);

}

int main(){

scanf("%d%d",&n,&m);

rep(i,1,n) scanf("%d",a+i);

rep(i,1,n) build(1,1,n,i,a[i]);

rep(i,1,m){

int f; scanf("%d",&f);

If (f==1)

scanf("%d%d%d",&x,&y,&k),tmp=1,getrank(1,1,n,x,y,k),printf("%d\n",tmp);

if (f==2) scanf("%d%d%d",&x,&y,&k),getidx(x,y,k);

if (f==3) scanf("%d%d",&x,&y),change(1,1,n,x,y,a[x]),a[x]=y;

if (f==4)

scanf("%d%d%d",&x,&y,&k),tmp=0,askpre(1,1,n,x,y,k),printf("%d\n",tmp);

if (f==5)

scanf("%d%d%d",&x,&y,&k),tmp=inf,asknxt(1,1,n,x,y,k),printf("%d\n",tmp);

}

return 0;

}

主席树：

//静态主席树

var

i,l,r,x,m,n,sz,tot:longint;

root,a,num,hash:array[0..100100] of longint;

lc,rc,s:array[0..2000100] of longint;

procedure qsort(l,r:longint);

var i,j,k,t:longint;

begin

i:=l; j:=r; k:=num[(l+r)>>1];

repeat

while num[i]<k do inc(i);

while num[j]>k do dec(j);

if i<=j then

begin

t:=num[i]; num[i]:=num[j]; num[j]:=t;

inc(i); dec(j);

end;

until i>j;

if l<j then qsort(l,j);

if i<r then qsort(i,r);

end;

function find(x:longint):longint;

var l,r,mid:longint;

begin

l:=1; r:=tot;

while l<=r do

begin

mid:=(l+r)>>1;

if hash[mid]<x then l:=mid+1 else r:=mid-1;

end;

exit(l);

end;

procedure insert(l,r,x:longint; var y:longint; v:longint);

var mid:longint;

begin

inc(sz); y:=sz; s[y]:=s[x]+1;

if l=r then exit;

lc[y]:=lc[x]; rc[y]:=rc[x];

mid:=(l+r)>>1;

if v<=mid then insert(l,mid,lc[x],lc[y],v)

else insert(mid+1,r,rc[x],rc[y],v);

end;

function ask(l,r,x,y,k:longint):longint;

var mid:longint;

begin

if l=r then exit(l);

mid:=(l+r)>>1;

if s[lc[y]]-s[lc[x]]>=k then exit(ask(l,mid,lc[x],lc[y],k))

else exit(ask(mid+1,r,rc[x],rc[y],k-(s[lc[y]]-s[lc[x]])));

end;

begin

readln(n,m);

for i:=1 to n do read(a[i]);

for i:=1 to n do num[i]:=a[i];

qsort(1,n); tot:=1; hash[tot]:=num[1];

for i:=2 to n do

if num[i]<>num[i-1] then

begin inc(tot); hash[tot]:=num[i]; end;

for i:=1 to n do insert(1,tot,root[i-1],root[i],find(a[i]));

for i:=1 to m do

begin

readln(l,r,x);

writeln(hash[ask(1,tot,root[l-1],root[r],x)]);

end;

end.

#include<cstdio>

#include<algorithm>

#include<cstring>

#define mid ((L+R)>>1)

#define mem(a) memset(a,0,sizeof(a))

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=100010,M=8000010;

char s[10];

int n,m,top,nd,l[30],r[30],a,b;

int T,v[N],num[N],f[N],A[N],B[N],K[N],rt[N],sum[M],ls[M],rs[M];

void upd(int y,int &x,int L,int R,int pos,int k){

x=++nd; sum[x]=sum[y]+k; ls[x]=ls[y]; rs[x]=rs[y];

if (L==R) return;

if (pos<=mid) upd(ls[x],ls[x],L,mid,pos,k); else upd(rs[x],rs[x],mid+1,R,pos,k);

}

int que(int L,int R,int k){

if (L==R) return L;

int sl=0,sr=0;

rep(i,1,a) sl+=sum[ls[l[i]]];

rep(i,1,b) sr+=sum[ls[r[i]]];

if (sr-sl>=k){

rep(i,1,a) l[i]=ls[l[i]];

rep(i,1,b) r[i]=ls[r[i]];

return que(L,mid,k);

}else{

rep(i,1,a) l[i]=rs[l[i]];

rep(i,1,b) r[i]=rs[r[i]];

return que(mid+1,R,k-(sr-sl));

}

}

int main(){

for (scanf("%d",&T); T--; ){

top=0; nd=0; mem(f); mem(ls); mem(rs);

scanf("%d%d",&n,&m);

rep(i,1,n) scanf("%d",v+i),num[++top]=v[i];

rep(i,1,m){

scanf("%s",s); scanf("%d%d",A+i,B+i);

if (s[0]=='Q') scanf("%d",K+i),f[i]=1; else num[++top]=B[i];

}

sort(num+1,num+top+1); top=unique(num+1,num+top+1)-num-1;

rep(i,1,n){

int t=lower\_bound(num+1,num+top+1,v[i])-num;

for (int j=i; j<=n; j+=j&-j) upd(rt[j],rt[j],1,top,t,1);

}

rep(i,1,m) if (f[i]){

a=0; b=0; A[i]--;

for (int j=A[i]; j; j-=j&-j) l[++a]=rt[j];

for (int j=B[i]; j; j-=j&-j) r[++b]=rt[j];

printf("%d\n",num[que(1,top,K[i])]);

}else{

int t=lower\_bound(num+1,num+top+1,v[A[i]])-num;

for (int j=A[i]; j<=n; j+=j&-j) upd(rt[j],rt[j],1,top,t,-1);

v[A[i]]=B[i]; t=lower\_bound(num+1,num+top+1,B[i])-num;

for (int j=A[i]; j<=n; j+=j&-j) upd(rt[j],rt[j],1,top,t,1);

}

}

return 0;

}

CDQ：

#include<cstdio>

#include<algorithm>

#define rep(i,l,r) for (int i=l; i<=r; i++)

using namespace std;

const int N=100100;

int n,m,x,y,ans,c[N],f[N];

struct P{ int a,mn,mx,x,y,id; }q[N],p[N];

bool cmp(P a,P b){ return (a.x==b.x)?(a.y==b.y)?a.id<b.id:a.y<b.y:a.x<b.x; }

void add(int x,int k){ for (; x<=100000; x+=x&-x) c[x]=k?max(c[x],k):0; }

int que(int x){ int res=0; for (; x; x-=x&-x) res=max(res,c[x]); return res; }

void solve(int l,int r){

if (l==r){ f[l]=max(f[l],1); return; }

int mid=(l+r)>>1; solve(l,mid);

rep(i,l,r)

if (q[i].id<=mid) p[i].x=q[i].a,p[i].y=q[i].mx,p[i].id=q[i].id;

else p[i].x=q[i].mn,p[i].y=q[i].a,p[i].id=q[i].id;

sort(p+l,p+r+1,cmp);

rep(i,l,r) if (p[i].id<=mid) add(p[i].y,f[p[i].id]); else f[p[i].id]=max(f[p[i].id],que(p[i].y)+1);

rep(i,l,r) if (p[i].id<=mid) add(p[i].y,0);

solve(mid+1,r);

}

int main(){

scanf("%d%d",&n,&m);

rep(i,1,n) scanf("%d",&q[i].a),q[i].mn=q[i].mx=q[i].a,q[i].id=i;

rep(i,1,m) scanf("%d%d",&x,&y),q[x].mn=min(q[x].mn,y),q[x].mx=max(q[x].mx,y);

solve(1,n);

rep(i,1,n) ans=max(ans,f[i]);

printf("%d\n",ans);

return 0;

}

整体二分：

type

query=record

x,y,k,s,opt,cur:longint;

end;

var

ch:char;

i,m,n,num,cnt,x,y,z:longint;

q,q1,q2:array[0..100100]of query;

a,ans,tmp,t:array[0..100100]of longint;

function lowbit(x:longint):longint;

begin

exit(x and (-x));

end;

procedure add(x,y:longint);

begin

while x<=n do

begin inc(t[x],y); inc(x,lowbit(x)); end;

end;

function ask(x:longint):longint;

var k:longint=0;

begin

while x>0 do

begin inc(k,t[x]); dec(x,lowbit(x)); end;

exit(k);

end;

procedure divide(head,tail,l,r:longint);

var i,mid,l1,l2:longint;

begin

if head>tail then exit;

if l=r then

begin

for i:=head to tail do

if q[i].opt=3 then ans[q[i].s]:=l;

exit;

end;

mid:=(l+r)>>1;

for i:=head to tail do

begin

if (q[i].opt=1)and(q[i].y<=mid) then add(q[i].x,1);

if (q[i].opt=2)and(q[i].y<=mid) then add(q[i].x,-1);

if q[i].opt=3 then tmp[i]:=ask(q[i].y)-ask(q[i].x-1);

end;

for i:=head to tail do

begin

if (q[i].opt=1)and(q[i].y<=mid) then add(q[i].x,-1);

if (q[i].opt=2)and(q[i].y<=mid) then add(q[i].x,1);

end;

l1:=0; l2:=0;

for i:=head to tail do

if q[i].opt=3 then

if q[i].cur+tmp[i]>=q[i].k then

begin inc(l1); q1[l1]:=q[i]; end

else

begin

inc(q[i].cur,tmp[i]); inc(l2); q2[l2]:=q[i];

end

else

if q[i].y<=mid then

begin inc(l1); q1[l1]:=q[i]; end

else

begin inc(l2); q2[l2]:=q[i]; end;

for i:=1 to l1 do q[head+i-1]:=q1[i];

for i:=1 to l2 do q[head+l1+i-1]:=q2[i];

divide(head,head+l1-1,l,mid);

divide(head+l1,tail,mid+1,r);

end;

begin

readln(n,m); cnt:=0; num:=0;

for i:=1 to n do

begin

read(a[i]); inc(num);

q[num].x:=i; q[num].y:=a[i];

q[num].opt:=1; q[num].s:=0;

end;

readln;

for i:=1 to m do

begin

read(ch);

if ch='Q' then

begin

read(ch); readln(x,y,z); inc(num);

q[num].x:=x; q[num].y:=y; q[num].k:=z;

q[num].opt:=3; inc(cnt); q[num].s:=cnt;

end

else

begin

read(ch); readln(x,y); inc(num);

q[num].x:=x; q[num].y:=a[x];

q[num].opt:=2; q[num].s:=0;

inc(num); q[num].x:=x; q[num].y:=y;

q[num].opt:=1; q[num].s:=0;

a[x]:=y;

end;

end;

divide(1,num,0,1000000000);

for i:=1 to cnt do writeln(ans[i]);

end.