1. Hdu6156(回文数位dp) （2017-8-22）

#include <iostream>

#include <cstring>

#include <cmath>

#include <queue>

#include <stack>

#include <list>

#include <map>

#include <set>

#include <string>

#include <cstdlib>

#include <cstdio>

#include <algorithm>

using namespace std;

int T;

typedef long long LL;

int a[33],t[33];

LL dp[33][33][2];

LL dfs(int len,int cur,int s,int fp,int y)

{

if(!cur)

return s?y:1;

if(!fp && dp[len][cur][s] != -1)

return dp[len][cur][s];

int n = fp?a[cur]:y-1;

LL res = 0;

for(int i=0;i<=n;i++)

{

t[cur] = i;

if(len == cur && i == 0)

res += dfs(len-1,cur-1,s,fp&&i==n,y);

else if(s && cur <= (len+1)/2)

res += dfs(len,cur-1,t[len-cur+1]==i,fp&&i==n,y);

else

res += dfs(len,cur-1,s,fp&&i==n,y);

}

if(!fp)

dp[len][cur][s] = res;

return res;

}

LL sum(LL x,LL y)

{

int len = 0;

while(x)

{

a[++len] = x % y;

x /= y;

}

//printf("len=%d\n",len);

return dfs(len,len,1,1,y);

}

const int N = 1e5+10;

LL L[N],R[N],l[N],r[N],ans[N];

int main()

{

int Ca=1;

scanf("%d",&T);

for (int ca=1;ca<=T;ca++){

scanf("%I64d%I64d%I64d%I64d",&L[ca],&R[ca],&l[ca],&r[ca]);

}

for (int i=2;i<=36;i++){

memset(dp,-1,sizeof dp);

for (int ca=1;ca<=T;ca++)if (l[ca]<=i && i<=r[ca]){

ans[ca]+=sum(R[ca],i)-sum(L[ca]-1,i);

}

}

for (int ca=1;ca<=T;ca++){

printf("Case #%d: %I64d\n",ca,ans[ca]);

}

return 0;

}

1. BZOJ2115最大xor路径（2017-8-22）

// <BZOJ2115.cpp> - Wed Aug 3 16:26:56 2016

// This file is created by XuYike's black technology automatically.

// Copyright (C) 2015 ChangJun High School, Inc.

// I don't know what this program is.

#include <iostream>

#include <vector>

#include <algorithm>

#include <cstring>

#include <cstdio>

#include <cmath>

using namespace std;

typedef long long lol;

int gi(){

int res=0,fh=1;char ch=getchar();

while((ch>'9'||ch<'0')&&ch!='-')ch=getchar();

if(ch=='-')fh=-1,ch=getchar();

while(ch>='0'&&ch<='9')res=res\*10+ch-'0',ch=getchar();

return fh\*res;

}

lol gl(){

lol res=0,fh=1;char ch=getchar();

while((ch>'9'||ch<'0')&&ch!='-')ch=getchar();

if(ch=='-')fh=-1,ch=getchar();

while(ch>='0'&&ch<='9')res=res\*10+ch-'0',ch=getchar();

return fh\*res;

}

const int MAXN=50010;

const int MAXM=200010;

const int INF=1e9;

int bt=1,b[MAXN],next[MAXM],to[MAXM];lol val[MAXM];

inline void add(lol z,int y,int x){

next[++bt]=b[x];b[x]=bt;to[bt]=y;val[bt]=z;

next[++bt]=b[y];b[y]=bt;to[bt]=x;val[bt]=z;

}

bool vis[MAXN];

lol dis[MAXN],h[MAXM];

int tot;

void dfs(int x,int f){

vis[x]=1;

for(int i=b[x];i;i=next[i]){

if((i^f)==1)continue;

if(vis[to[i]]){

if(i&1)h[++tot]=dis[x]^dis[to[i]]^val[i];

continue;

}

dis[to[i]]=dis[x]^val[i];

dfs(to[i],i);

}

}

int main(){

int n=gi(),m=gi();

for(int i=1;i<=m;i++)add(gl(),gi(),gi());

dfs(1,0);

lol ans=dis[n];

for(lol p=1ll<<62;p;p>>=1){

int fd=0;

for(int i=1;i<=tot;i++)

if(h[i]&p)

if(fd)h[i]^=h[fd];

else fd=i;

if(fd&&!(ans&p))ans^=h[fd];

h[fd]=0;

}

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

return 0;

}

/\*

author : VFVrPQ

problem : 2017MUTC9-1006

solve : 枚举二进制位，把当前为0的和当前为1的分成两个集合,求一次最短路

time : 2017-08-23-18.04.34

\*/

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

const int N = 1e5+10;

const int M = 1e9+7;

const LL INF = 1LL<<60;

vector< pair<int,int> > V[N];

int n,m;

LL d[N];

int vis[N];

int K;

int a[N];

int S,T;

LL dijkstra(int S,int T)

{

//printf("T=%d\n",T);

for (int i=0;i<n+2;i++) d[i]=INF,vis[i]=0;

priority\_queue< pair<LL,int> > Q;

d[S]=0;

Q.emplace(0LL,S);

while (!Q.empty()){

int u = Q.top().second; Q.pop();

if (vis[u])continue;

vis[u]=1;

for (auto pa:V[u]){

if (d[pa.first]>d[u]+pa.second){

d[pa.first]=d[u]+pa.second;

if (!vis[pa.first]){

Q.emplace(-d[pa.first],pa.first);

}

}

}

}

//printf("%d,%d,d[T]=%lld\n",S,T,d[T]);

return d[T];

}

void Delete(int T,int bit,int fz){

for (int i=0;i<K;i++)if (!(a[i]>>bit&1^fz)){

pair<int,int> x = make\_pair(T,0);

for (vector< pair<int,int> >::iterator it=V[a[i]].begin();it != V[a[i]].end();it++){

if (\*it == x) {

V[a[i]].erase(it);

break;

}

}

}

}

int main()

{

//freopen("1.txt","r",stdin);

int Case;scanf("%d",&Case);

for (int Ca=1;Ca<=Case;Ca++)

{

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

for (int i=0;i<n;i++)V[i].clear();

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

int u,v,w;scanf("%d%d%d",&u,&v,&w);u--;v--;

V[u].emplace\_back(v,w);

}

scanf("%d",&K);

for (int i=0;i<K;i++) scanf("%d",&a[i]),a[i]--;

S=n;T=n+1;

LL ans=INF;

for (int bit=20;bit>=0;bit--){

for (int fz=0;fz<2;fz++){

V[S].clear();

V[T].clear();

for (int i=0;i<K;i++)if ((a[i]>>bit&1)^fz){

V[S].emplace\_back(a[i],0);

}else{

V[a[i]].emplace\_back(T,0);

}

if (V[S].size()==K || V[T].size()==K) {

}else{

ans=min(ans,dijkstra(S,T));

}

//if (dijkstra(S,T)==0) printf("ans=0,bit=%d\n",bit);

Delete(T,bit,fz);

}

}

printf("Case #%d: %lld\n",Ca,ans);

}

return 0;

}

/\*

author : VFVrPQ

problem : 2017MUTC10-1008 MONKEYS

solve : 读入优化（fread(buf, 1, BUF\_SIZE-1, stdin);）；增加一条边可以覆盖两只猴子较优，找出覆盖两只猴子的最大边数，剩下的增加一条边只增加一个点(练习的时候fread无法stdin输出，没调试，后来发现u用scanf读入QAQ，改成getInt就AC了）

time : 2017-08-24-17.09.07

\*/

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

const int M = 1e9+7;

const int N = 100000+10;

const LL INF = 1LL<<60;

const int BUF\_SIZE = 100\*1024\*1024;

char buf[BUF\_SIZE];

int buf\_h=0;

vector<int> V[N];

int f[N][2];

int n,K;

int getInt()

{

char c = buf[buf\_h++];int x=0;

while (c>'9' || c<'0') c=buf[buf\_h++];

while (1){

x=x\*10+c-'0';

c=buf[buf\_h++];

if (c>'9' || c<'0') return x;

}

}

void dfs(int u,int fa=-1)

{

for (auto v:V[u]){

if (v==fa) continue;

dfs(v,u);

f[u][0]+=f[v][1];

}

for (auto v:V[u]){

if (v==fa) continue;

f[u][1] = max(f[u][1], f[u][0]-f[v][1]+f[v][0]+2);

}

}

int main()

{

//FILE \* fp = fopen("1.txt","r");

fread(buf, 1, BUF\_SIZE-1, stdin);

//fclose(fp);

//printf("buf=%s",buf);

//printf("%d\n",strlen(buf));

int Case = getInt();

while (Case--){

n = getInt(); K = getInt();

for (int i=0;i<n;i++) V[i].clear();

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

int u;u=getInt();u--;

V[u].push\_back(i+1);

V[i+1].push\_back(u);

}

memset(f,0,sizeof f);

dfs(0);

int ans = max(f[0][0],f[0][1]);

if (ans>=K) ans = (K+1)/2;

else {

ans=ans/2+K-ans;

}

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

}

return 0;

}

/\*

author : VFVrPQ

problem : cf844-D ,交互题（第一次做）。递增单向链表（n<=50000)，问>=x的值。每次询问返回值和下个位置的索引，最多只能询问2000次

solve : 玄学啊，询问 start和随机999个数，再从<x的最大的数出发，错误的概率是（1-999/n)^1000=1.7\*(10^-9);我在比赛的时候n=1没考虑；订正的时候srand没加！！

time : 2017-08-25-07.59.18

\*/

#include<bits/stdc++.h>

using namespace std;

typedef long long LL;

const int M = 1e9+7;

const int N = 50000+10;

int ans;

int n,start,x;

int perm[N];

int a[N],Next[N];

void query(int u)

{

if (a[u]>=0) return ;

printf("? %d\n",u);

fflush(stdout);

scanf("%d%d",&a[u],&Next[u]);

}

int main(){

srand(unsigned(time(NULL)));//这句话一定要加

memset(a,-1,sizeof a);

scanf("%d%d%d",&n,&start,&x);

query(start);

for (int i=0;i<n;i++) perm[i]=i+1;

random\_shuffle(perm,perm+n);

for (int i=0;i<999 && i<n;i++){

int u = perm[i];

query(u);

if (a[u]<x && a[u]>a[start])

start = u;

}

//puts("ok");

while (a[start]<x){

start = Next[start];

if (start==-1) break;

query(start);

}

printf("! %d\n", (start==-1)?-1:a[start]);

return 0;

}

1. 树的重心（n为奇数只有1个，n为偶数有两个）

int c1=-1,c2=-1;

int dfs(int u,int p=-1){

int s=1;

for (auto v:V[u])if (v^p){

s+=dfs(v,u);

}

if (c1 == -1 && 2\*s>=n){

c1=u;

if (s\*2==n) c2=p;

else c2=-1;

}

return s;

}

1. 线性递推求任意项（2017MUTC10-2），输入前几项和n就可以得出。

#include <bits/stdc++.h>

using namespace std;

#define rep(i,a,n) for (int i=a;i<n;i++)

#define per(i,a,n) for (int i=n-1;i>=a;i--)

#define pb push\_back

#define mp make\_pair

#define all(x) (x).begin(),(x).end()

#define fi first

#define se second

#define SZ(x) ((int)(x).size())

typedef vector<int> VI;

typedef long long ll;

typedef pair<int,int> PII;

const ll mod=1000000007;

ll powmod(ll a,ll b) {ll res=1;a%=mod; assert(b>=0); for(;b;b>>=1){if(b&1)res=res\*a%mod;a=a\*a%mod;}return res;}

int \_;

ll n;

namespace linear\_seq {

const int N=10010;

ll res[N],base[N],\_c[N],\_md[N];

vector<int> Md;

void mul(ll \*a,ll \*b,int k) {

rep(i,0,k+k) \_c[i]=0;

rep(i,0,k) if (a[i]) rep(j,0,k) \_c[i+j]=(\_c[i+j]+a[i]\*b[j])%mod;

for (int i=k+k-1;i>=k;i--) if (\_c[i])

rep(j,0,SZ(Md)) \_c[i-k+Md[j]]=(\_c[i-k+Md[j]]-\_c[i]\*\_md[Md[j]])%mod;

rep(i,0,k) a[i]=\_c[i];

}

int solve(ll n,VI a,VI b) { // a 系数 b 初值 b[n+1]=a[0]\*b[n]+...

// printf("%d\n",SZ(b));

ll ans=0,pnt=0;

int k=SZ(a);

assert(SZ(a)==SZ(b));

rep(i,0,k) \_md[k-1-i]=-a[i];\_md[k]=1;

Md.clear();

rep(i,0,k) if (\_md[i]!=0) Md.push\_back(i);

rep(i,0,k) res[i]=base[i]=0;

res[0]=1;

while ((1ll<<pnt)<=n) pnt++;

for (int p=pnt;p>=0;p--) {

mul(res,res,k);

if ((n>>p)&1) {

for (int i=k-1;i>=0;i--) res[i+1]=res[i];res[0]=0;

rep(j,0,SZ(Md)) res[Md[j]]=(res[Md[j]]-res[k]\*\_md[Md[j]])%mod;

}

}

rep(i,0,k) ans=(ans+res[i]\*b[i])%mod;

if (ans<0) ans+=mod;

return ans;

}

VI BM(VI s) {

VI C(1,1),B(1,1);

int L=0,m=1,b=1;

rep(n,0,SZ(s)) {

ll d=0;

rep(i,0,L+1) d=(d+(ll)C[i]\*s[n-i])%mod;

if (d==0) ++m;

else if (2\*L<=n) {

VI T=C;

ll c=mod-d\*powmod(b,mod-2)%mod;

while (SZ(C)<SZ(B)+m) C.pb(0);

rep(i,0,SZ(B)) C[i+m]=(C[i+m]+c\*B[i])%mod;

L=n+1-L; B=T; b=d; m=1;

} else {

ll c=mod-d\*powmod(b,mod-2)%mod;

while (SZ(C)<SZ(B)+m) C.pb(0);

rep(i,0,SZ(B)) C[i+m]=(C[i+m]+c\*B[i])%mod;

++m;

}

}

return C;

}

int gao(VI a,ll n) {

VI c=BM(a);

c.erase(c.begin());

rep(i,0,SZ(c)) c[i]=(mod-c[i])%mod;

return solve(n,c,VI(a.begin(),a.begin()+SZ(c)));

}

};

void ptr()

{

ll h[10], b[10];

h[0] = 2, h[1] = 3, h[2] = 6;

for(int i = 3;i <= 10;i ++) h[i] = 4 \* h[i - 1] + 17 \* h[i - 2] - 12 \* h[i - 3] - 16;

for(int i = 1;i <= 10;i ++) b[i] = 3 \* h[i + 1] \* h[i] + 9 \* h[i + 1] \* h[i - 1] + 9 \* h[i] \* h[i] + 27 \* h[i] \* h[i - 1] - 18 \* h[i + 1] - 126 \* h[i] - 81 \* h[i - 1] + 192;

for(int i = 2;i <= 8;i ++) cout << fixed << setprecision(0) << floor(sqrt(b[i] + (1LL<<2 \* i))) << endl;

}

int main() {

int T;

//ptr();

scanf("%d", &T);

while(T --) {

scanf("%lld",&n);

printf("%lld\n",linear\_seq::gao(VI{31,197,1255,7997},n - 2));

}

return 0;

}

1. Cf849E

/\*

author : VFVrPQ

problem : cf849-E ,Difference between last occurrence and first occurrence equals the sum of differences between pairs of adjacent occurrences. Handle this with some segment tree

给定一个数组n，和若干询问Q, (n,Q<=1e5, a[i]<=1e5)

1.修改位置的值，变成x(1<=x<=n);

2.询问区间内 同一个值第一次出现和最后一次出现的差值 的和。

solve : 线段树套树状数组（第一次写，改了好久QAQ）。首先，对于同一个值在区间内的差值可以转化为相邻位置的差值之和。所以对于询问[l,r]，答案是sigma{i-pre[i]},其中pre[i]>=l。

建一棵序号线段树，每个节点建一棵数值树状数组；每次询问 树状数组给出val>=的值.需要用数据结构维护一下额外的东西

time : 2017-09-03-13.35.29

\*/

#include <bits/stdc++.h>

using namespace std;

//#define DEBUG

const int N = 100000+10;

typedef long long LL;

struct Query{

int t,l,r;

Query(int t=0,int l=0,int r=0):t(t),l(l),r(r){}

};

vector<Query> Que;

int n,Q;

int pr[N];

int a[N],aa[N];

int ls[N];

vector<int> vars[N\*4];

vector<LL> T[N\*4];

set<int> ids[N\*4];

LL Tsum[N\*4];

#ifdef DEBUG

int maxp = 0;

#endif // DEBUG

void addSTF(int p,int pos,int n,int val){

Tsum[p]+=val;

for (;pos<=n;pos+=pos&(-pos))

T[p][pos]+=val;

}

void build(int k,int p,int l,int r)

{

#ifdef DEBUG

maxp = max(maxp,p);

#endif // DEBUG

for (int i=l;i<=r;i++){

if (k==0){

vars[p].push\_back(pr[i]);

}else{

int pos = int(lower\_bound(vars[p].begin(),vars[p].end(),pr[i])-vars[p].begin());

assert(pos != vars[p].size());

pos++;

addSTF(p,pos,int(vars[p].size()),i-pr[i]);

}

}

if (l == r) return ;

int mid=(l+r)>>1;

build (k,p<<1,l,mid);

build (k,p<<1|1,mid+1,r);

}

void updateST(int k,int p,int l,int r,int pos,int pos2,int val){

if (k==0){

vars[p].push\_back(pos2);

}else{

int mypos = int(lower\_bound(vars[p].begin(),vars[p].end(),pos2)-vars[p].begin());

assert(mypos != vars[p].size());

mypos++;

addSTF(p,mypos,int(vars[p].size()),val);

}

if (l==r) return ;

int m = (l+r)>>1;

if (pos<=m) updateST(k,p<<1,l,m,pos,pos2,val);

else updateST(k,p<<1|1,m+1,r,pos,pos2,val);

}

void DeleteST(int k,int l){

updateST(k,1,1,n,l,pr[l],-(l-pr[l]) );

auto it\_np = ids[ a[l] ].upper\_bound(l);

if (it\_np != ids[ a[l] ].end()){

int np = \*it\_np;

updateST(k,1,1,n,np,pr[np],-(np-pr[np]));

pr[np] = pr[l];

updateST(k,1,1,n,np,pr[np],np-pr[np]);

}

ids[ a[l] ].erase(l);

pr[l]=0;

a[l]=0;

}

void InsertST(int k,int l,int val){

auto it\_np = ids[val].upper\_bound(l);

int pre = 0;

if (it\_np != ids[val].end()){

int np = \*it\_np;

updateST(k,1,1,n,np,pr[np],-(np-pr[np]));

pre = pr[np];

pr[np] = l;

updateST(k,1,1,n,np,pr[np],(np-pr[np]));

}else

if (it\_np != ids[val].begin()){

it\_np--;

pre = \*it\_np;

}

pr[l] = pre;

a[l] = val;

updateST(k,1,1,n,l,pr[l],(l-pr[l]));

ids[val].insert(l);

}

LL getSumSTF(int p,int pos){

LL ret=0;

for (;pos;pos-=pos&(-pos))

ret+=T[p][pos];

return ret;

}

LL getSum(int p,int l,int r,int L,int R,int myL){

if (l==L && R==r){

int pos = int(lower\_bound(vars[p].begin(),vars[p].end(),myL) - vars[p].begin());

return Tsum[p]-getSumSTF(p,pos);

}

int mid=(l+r)>>1;

LL ret=0;

if (L<=mid)

ret += getSum(p<<1,l,mid,L,min(mid,R),myL);

if (mid< R)

ret += getSum(p<<1|1,mid+1,r,max(L,mid+1),R,myL);

return ret;

}

int main()

{

#ifdef DEBUG

freopen("1.txt","r",stdin);

#endif // DEBUG

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

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

scanf("%d",&aa[i]);

}

for (int i=1;i<=Q;i++){

int t,l,r;scanf("%d%d%d",&t,&l,&r);

Que.emplace\_back(t,l,r);

}

for (int k=0;k<2;k++){

if (k){//initial

for (int i=1;i<=n;i++) ids[i].clear(),ls[i]=0;

for (int i=0;i<4\*N;i++)if (vars[i].size()>=1){

sort(vars[i].begin(),vars[i].end());

vars[i].erase(unique(vars[i].begin(),vars[i].end()),vars[i].end());

T[i].assign(vars[i].size()+1,0);

}

#ifdef DEBUG

// printf("maxp=%d\n",maxp);

// for (int i=1;i<=maxp;i++){

// printf("%d:",i);

// for (auto var:vars[i]){

// printf("%d ",var);

// }

// puts("");

// }

#endif // DEBUG

}

for (int i=1;i<=n;i++) a[i]=aa[i];

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

pr[i] = ls[ a[i] ];

ls[ a[i] ] =i;

ids[a[i]].insert(i);

}

build(k,1,1,n);//ok

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

int t,l,r; t=Que[i].t;l=Que[i].l;r=Que[i].r;

if (t==1){

DeleteST(k,l);

InsertST(k,l,r);

}else{

if (k) printf("%I64d\n",getSum(1,1,n,l,r,l));

}

}

}

return 0;

}

1. Cf854-E

/\*

author : VFVrPQ

problem : cf854-E,n\*n的网格有n个点，横坐标从1->n,纵坐标题目输入（也是1-n），n个点各不相同。

定义beautiful rectangle为矩阵的两个对角分别有两个点(共n\*(n-1)/2个矩阵）。

每次给出一个矩阵左下角(l,d),右上角(r,u)，询问与多少个beautiful rectangle相交

n,q<=2e5

solve : 直接求比较难，考虑总的-不成立的。总的是n\*(n-1)/2,不成立的是四个边-四个小角。每次用主席树求解横坐标[l,r]，纵坐标[d,u]内点的个数。

time : 2017-09-08-21.04.19

\*/

#include<bits/stdc++.h>

using namespace std;

typedef long long LL;

const int M = 1e9+7;

const int N = 2e5+10;

//#define DEBUG

int p[N];

int root[N];

struct Tree{

LL num;

int l,r;

Tree(){}

Tree(LL num):num(num){}

};

Tree tr[N\*20];

int cnt;

void Insert(int&root,int l,int r,int pos,LL num){

cnt++;

tr[cnt]=tr[root];

root=cnt;

if (l==r){

tr[root].num+=num;

return ;

}

int m = (l+r)>>1;

if (pos<=m) Insert(tr[root].l,l,m,pos,num);

else Insert(tr[root].r,m+1,r,pos,num);

tr[root].num = tr[tr[root].l].num+tr[tr[root].r].num;

}

LL Find2(int root,int l,int r,int L,int R){

if (l==L && r==R){

return tr[root].num;

}

int m=(l+r)>>1;

if (R<=m) return Find2(tr[root].l,l,m,L,R);

else if (m< L) return Find2(tr[root].r,m+1,r,L,R);

else{

Tree ret1 = Find2(tr[root].l,l,m,L,m);

Tree ret2 = Find2(tr[root].r,m+1,r,m+1,R);

return ret1.num+ret2.num;

}

}

int n,Q;

LL Find(int root1,int root2,int l,int r){

if (l>r) return 0;

LL ans1 = Find2(root1,1,n,l,r);

LL ans2 = Find2(root2,1,n,l,r);

return (ans2-ans1)\*(ans2-ans1-1)/2;

}

int main(){

#ifdef DEBUG

freopen("1.txt","r",stdin);

#endif // DEBUG

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

for (int i=1;i<=n;i++)scanf("%d",&p[i]);

//zhuxishu

root[0]=0;cnt=0;

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

root[i]=root[i-1];

Insert(root[i],1,n,p[i],1);

}

LL tot = Find(root[0],root[n],1,n);

while (Q--){

int l,d,r,u;scanf("%d%d%d%d",&l,&d,&r,&u);

LL ans = Find(root[0],root[l-1],1,n);

ans=ans+Find(root[r],root[n],1,n);

ans=ans+Find(root[0],root[n],u+1,n);

ans=ans+Find(root[0],root[n],1,d-1);

ans=ans-Find(root[0],root[l-1],1,d-1);

ans=ans-Find(root[0],root[l-1],u+1,n);

ans=ans-Find(root[r],root[n],1,d-1);

ans=ans-Find(root[r],root[n],u+1,n);

printf("%I64d\n",tot-ans);

}

return 0;

}