**Perfect Header**

**#pragma comment(linker, "/stack:200000000")**

**#pragma GCC optimize("Ofast")**

**#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,tune=native")**

**#pragma GCC optimize("unroll-loops")**

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

**#include<ext/pb\_ds/assoc\_container.hpp>**

**#include<ext/pb\_ds/tree\_policy.hpp>**

**using namespace \_\_gnu\_pbds;**

**using namespace std;**

**#define ll long long**

**#define ull unsigned long long**

**#define ld long double**

**#define pii pair<int,int>**

**#define pll pair<ll,ll>**

**#define vi vector<int>**

**#define vll vector<ll>**

**#define vc vector<char>**

**#define vs vector<string>**

**#define vpii vector< pair<int,int> >**

**#define vpll vector< pair<ll,ll> >**

**#define ppll pair< ll,pll >**

**#define pllp pair< pll,ll >**

**#define stll stack<ll>**

**#define qll queue<ll>**

**#define pqll priority\_queue<ll>**

**#define mppll map<pll,ll>**

**#define mpii map<int,int>**

**#define mpll map<ll,ll>**

**#define mpss map<string,string>**

**#define mpsll map<string,ll>**

**#define mplls map<ll,string>**

**#define umpii unordered\_map<int,int>**

**#define umll unordered\_map<ll,ll>**

**#define umpss unordered\_map<string,string>**

**#define umpsll unordered\_map<string,ll>**

**#define umplls unordered\_map<ll,string>**

**#define umap unordered\_map**

**#define uset unordered\_set**

**#define PQ priority\_queue**

**#define rep(i,n) for(i=0;i<n;i++)**

**#define itfor(i, c) for (typeof((c).begin()) i = (c).begin(); i != (c).end(); i++)**

**#define printa(a,L,R) for(int i=L;i<R;i++) cout<<a[i]<<(i==R-1?'\n':' ')**

**#define printv(a) printa(a,0,a.size())**

**#define print2d(a,r,c) for(int i=0;i<r;i++) for(int j=0;j<c;j++) cout<<a[i][j]<<(j==c-1?'\n':' ')**

**#define pb push\_back**

**#define MP make\_pair**

**#define UB upper\_bound**

**#define LB lower\_bound**

**#define SQ(x) ((x)\*(x))**

**#define issq(x) (((ll)(sqrt((x))))\*((ll)(sqrt((x))))==(x))**

**#define F first**

**#define S second**

**#define mem(a,x) memset(a,x,sizeof(a))**

**#define inf 0x3f3f3f3f**

**#define PI 3.14159265358979323846**

**#define E 2.71828182845904523536**

**#define gamma 0.5772156649**

**#define nl "\n"**

**#define lg(r,n) (int)(log2(n)/log2(r))**

**#define sf(a) scanf("%d",&a)**

**#define sfl(a) scanf("%I64d",&a)**

**#define sfc(a) scanf("%c",&a)**

**#define sff(a,b) scanf("%d %d",&a,&b)**

**#define sffl(a,b) scanf("%I64d %I64d",&a,&b)**

**#define sfff(a,b,c) scanf("%d %d %d",&a,&b,&c)**

**#define sfffl(a,b,c) scanf("%I64d %I64d %I64d",&a,&b,&c)**

**#define pf printf**

**#define pfi(a) pf("%d\n",&a)**

**#define pfl(a) pf("%I64d\n",&a)**

**#define \_ccase printf("Case %d: ",++cs)**

**#define \_case cout<<"Case "<<++cs<<": "**

**#define debug(x) cout<<#x"="<<(x)<<nl**

**#define rev(v) reverse(v.begin(),v.end())**

**#define srt(v) sort(v.begin(),v.end())**

**#define grtsrt(v) sort(v.begin(),v.end(),greater<int>())**

**#define all(v) v.begin(),v.end()**

**#define mnv(v) \*min\_element(v.begin(),v.end())**

**#define mxv(v) \*max\_element(v.begin(),v.end())**

**#define countv(v,a) count(v.begin(),v.end(),a)**

**#define toint(a) atoi(a.c\_str())**

**#define midlr mid=b+((e-b)>>1),l=n<<1,r=n<<1|1**

**#define fast ios\_base::sync\_with\_stdio(false),cin.tie(NULL)**

**string tostr(int n) {stringstream rr;rr<<n;return rr.str();}**

**template <typename T> using ordered\_set = tree<T, null\_type, less<T>, rb\_tree\_tag, tree\_order\_statistics\_node\_update>;**

**const int mod=1e9+7;**

**//ll qpow(ll n,ll k) {ll ans=1;n%=mod;while(k){if(k&1) ans=(ans\*n)%mod;n=(n\*n)%mod;k>>=1;}return ans%mod;}**

**const int mxn=1e5+9;**

**const ld eps=1e-9;**

**int main()**

**{**

**fast;**

**return 0;**

**}**

**Closest Pair Points**

const int mod=1e9+7;

const int mxn=1e5+9;

const ld eps=1e-9;

//ll qpow(ll n,ll k) {ll ans=1;while(k){if(k&1) ans=(ans\*n)%mod;n=(n\*n)%mod;k>>=1;}return ans;}

struct point

{

ld x,y;

}px[mxn],py[mxn];

bool cmpx(point a,point b)

{

return (a.x==b.x?a.y<b.y:a.x<b.x);

}

bool cmpy(point a,point b)

{

return (a.y==b.y?a.x<b.x:a.y<b.y);

}

ld dis(point a,point b){ return sqrt(SQ(a.x-b.x)+SQ(a.y-b.y));}

ld bruteforce(point p[],ll n)

{

ll i,j;

ld d=inf;

for(i=0;i<n;i++) for(j=i+1;j<n;j++) d=fmin(d,dis(p[i],p[j]));

return d;

}

ld closest(point p[],ll n,ld mn)

{

ll i,j;

ld d=inf;

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

for(j=i+1;j<n&&abs(p[j].y-p[i].y)<mn;j++){

d=fmin(d,dis(p[i],p[j]));

}

}

return d;

}

ld yo(point px[],point py[],ll n)

{

if(n<=3) return bruteforce(px,n);

ll mid,i,j=0,l=0,r=0;

point pyl[n],pyr[n];

ld dl,dr,d;

mid=n/2;

point midp=px[mid];

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

if(py[i].x<=midp.x) pyl[l++]=py[i];

else pyr[r++]=py[i];

}

dl=yo(px,pyl,mid);

dr=yo(px+mid,pyr,n-mid);

d=fmin(dl,dr);

point tmp[n];

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

if(abs(py[i].x-midp.x)<d) tmp[j++]=py[i];

}

return fmin(d,closest(tmp,j,d));

}

int main()

{

fast;

ll i,j,k,n,m;

cin>>n;

for(i=0;i<n;i++) cin>>px[i].x>>px[i].y,py[i]=px[i];

sort(px,px+n,cmpx);

sort(py,py+n,cmpy);

cout<<fixed<<setprecision(10)<<yo(px,py,n)<<nl;

return 0;

}

**Divisors in O(n1/3)**

const int mod=1e9+7;

const int mxn=1e6+9;

const ld eps=1e-9;

//ll qpow(ll n,ll k) {ll ans=1;while(k){if(k&1) ans=(ans\*n)%mod;n=(n\*n)%mod;k>>=1;}return ans;}

vector<ull> v;

bool ck[mxn];

ull mulmod(ull a,ull b, ull mod)

{

ull ans=0;

a=a%mod;

while(b){

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

a=(a\*2)%mod;

b>>=1;

}

return ans;

}

ull qpow(ull n,ull k,ull mod)

{

ull ans=1;

n=n%mod;

while(k){

if(k&1) ans=mulmod(ans,n,mod);

n=mulmod(n,n,mod);

k>>=1;

}

return ans;

}

bool test(ull n,ull d)

{

ull i,j,k,a,x;

a=2+rand()%(n-4);

x=qpow(a,d,n);

if(x==1||x==n-1) return 1;

while(d!=n-1){

x=mulmod(x,x,n);

d\*=2;

if(x==1) return 0;

if(x==n-1) return 1;

}

return 0;

}

bool millerrobin(ull n)

{

if(n<=1||n==4) return 0;

if(n<=3) return 1;

ull i,d=n-1;

while(d%2==0) d/=2;

for(i=1;i<=20;i++) if(test(n,d)==0) return 0;

return 1;

}

void sieve()

{

ull i,j;

for(i=3;i\*i<mxn;i+=2) if(ck[i]==0) for(j=i\*i;j<mxn;j+=2\*i) ck[j]=1;

v.pb(2);

for(i=3;i<mxn;i+=2) if(ck[i]==0) v.pb(i);

}

bool issqr(ull n)

{

ull k=sqrt(n);

return k\*k==n;

}

ull countdiv(ull n)

{

ull i,cnt=0,ans=1;

for(i=0;;i++){

if(v[i]\*v[i]\*v[i]>n) break;

cnt=0;

while(n%v[i]==0){

n/=v[i];

cnt++;

}

ans\*=(cnt+1);

}

if(millerrobin(n)) ans\*=2;

else if(issqr(n)&&millerrobin((ull)sqrt(n))) ans\*=3;

else if(n!=1) ans\*=4;

return ans;

}

int main()

{

fast;

ull n;

sieve();

cin>>n;

cout<<countdiv(n);

return 0;

}

**Second Shortest Path**

vpii g[mxn];

int d[mxn][2];

int dijkstra(int s,int e)

{

priority\_queue< ppi,vector<ppi>,greater<ppi> > q;

int i,j,k,tmp,tmpp,x,w,v,fbest,sbest,p,qq;

si st;

d[s][0]=0;

q.push(mp(mp(0,inf),s));

while(!q.empty()){

ppi tp=q.top();

q.pop();

fbest=tp.fi.fi;

sbest=tp.fi.se;

v=tp.se;

for(i=0;i<g[v].size();i++){

x=g[v][i].fi;

w=g[v][i].se;

st.clear();

tmp=min(inf,fbest+w);

tmpp=min(inf,sbest+w);

st.insert(tmp);st.insert(tmpp);st.insert(d[x][0]);st.insert(d[x][1]);

sit it=st.begin();

p=\*it++;qq=\*it;

if(p<d[x][0]||qq<d[x][1]){

d[x][0]=p;

d[x][1]=qq;

q.push(mp(mp(d[x][0],d[x][1]),x));

}

}

}

return d[e][1];

}

int main()

{

int n, u, v, w, e, i, ret, t, cs = 1;

scanf("%d", &t);

while(t--) {

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

for(i = 1; i <= n; i++) g[i].clear();

memset(d, inf, sizeof d);

for(i = 0; i < e; i++) {

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

g[u].pb(mp(v,w));

g[v].pb(mp(u,w));

}

ret = dijkstra(1, n);

if(ret ==inf) printf("Case %d: -1\n", cs++);

else printf("Case %d: %d\n", cs++, ret);

}

return 0;

}

**SCC Kosaraju’s Algorithm**

const int mod=1e9+7;

const int mxn=1e5+9;

const ld eps=1e-9;

ll qpow(ll n,ll k) {ll ans=1;while(k){if(k&1) ans=(ans\*n)%mod;n=(n\*n)%mod;k>>=1;}return ans;}

vll g[mxn],grev[mxn];

bool vis[mxn];

stack<ll>st;

void dfs(ll i)

{

vis[i]=1;

for(auto x:g[i]) if(!vis[x]) dfs(x);

st.push(i);

}

void dfsrev(ll i)

{

vis[i]=1;

for(auto x:grev[i]) if(!vis[x]) dfsrev(x);

}

int main()

{

fast;

ll i,j,k,n,m,x,y,ans=0;

cin>>n>>m;

for(i=0;i<m;i++) cin>>x>>y,g[x].pb(y),grev[y].pb(x);

for(i=1;i<=n;i++) if(!vis[i]) dfs(i);

mem(vis,0);

while(!st.empty()){

i=st.top();

st.pop();

if(!vis[i]) dfsrev(i),ans++;

}

cout<<ans<<nl;

return 0;

}

**Max Flow**

const int mod=1e9+7;

const int mxn=1e5+9;

const ld eps=1e-9;

ll qpow(ll n,ll k) {ll ans=1;while(k){if(k&1) ans=(ans\*n)%mod;n=(n\*n)%mod;k>>=1;}return ans;}

vll g[5100];

ll res[5100][5100],s,des,vis[5100],par[5100];

bool bfs()

{

ll i,j,k,u;

mem(vis,0);

mem(par,0);

queue<ll>q;

q.push(s);

vis[s]=1;

while(!q.empty()){

u=q.front();

q.pop();

for(auto v:g[u]){

if(!vis[v]&&res[u][v]>0){

par[v]=u;

vis[v]=1;

q.push(v);

}

}

}

return vis[des]==1;

}

ll ford()

{

ll i,j,k,p,flow,ans=0;

while(bfs()){

flow=inf,k=des;

while(k!=s){

p=par[k];

flow=min(flow,res[p][k]);

k=p;

}

k=des;

while(k!=s){

p=par[k];

res[p][k]-=flow;

res[k][p]+=flow;

k=p;

}

ans+=flow;

}

return ans;

}

int main()

{

fast;

ll i,j,k,n,m,x,y,w;

cin>>n>>m;

for(i=0;i<m;i++) cin>>x>>y>>w,g[x].pb(y),res[x][y]+=w,res[y][x]+=w;

cin>>s>>des;

cout<< ford();

return 0;

}

**Maximum Bipartite Checking**

const int mod=1e9+7;

const int mxn=1e5+9;

const ld eps=1e-9;

ll qpow(ll n,ll k) {ll ans=1;while(k){if(k&1) ans=(ans\*n)%mod;n=(n\*n)%mod;k>>=1;}return ans;}

vll g[5100];

ll res[5100][5100],s,des,vis[5100],par[5100];

bool bfs()

{

ll i,j,k,u;

mem(vis,0);

mem(par,0);

queue<ll>q;

q.push(s);

vis[s]=1;

while(!q.empty()){

u=q.front();

q.pop();

for(auto v:g[u]){

if(!vis[v]&&res[u][v]>0){

par[v]=u;

vis[v]=1;

q.push(v);

}

}

}

return vis[des]==1;

}

ll ford()

{

ll i,j,k,p,flow,ans=0;

while(bfs()){

flow=inf,k=des;

while(k!=s){

p=par[k];

flow=min(flow,res[p][k]);

k=p;

}

k=des;

while(k!=s){

p=par[k];

res[p][k]-=flow;

res[k][p]+=flow;

k=p;

}

ans+=flow;

}

return ans;

}

int main()

{

fast;

ll i,j,k,n,m,x,y,w,e;

cin>>n>>m>>e;

for(i=0;i<e;i++) cin>>x>>y,g[x].pb(y+n),g[y+n].pb(x),res[x][y+n]++,res[y+n][x]++;

for(i=1;i<=n;i++) g[0].pb(i),g[i].pb(0),res[0][i]++,res[i][0]++;

for(i=1;i<=m;i++) g[i+n].pb(n+m+1),g[n+m+1].pb(i+n),res[i+n][n+m+1]++,res[n+m+1][i+n]++;

s=0,des=n+m+1;

cout<< ford();

return 0;

}

**KMP**

#include<bits/stdc++.h>

using namespace std;

int lps[1001000];

void pre\_kmp(string s)

{

int i,j,n;

i=1;

j=0;

lps[0]=0;

n=s.length();

while(i<n){

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

j++;

lps[i]=j;

i++;

}

else{

if(j==0){

lps[i]=0;

i++;

}

else{

j=lps[j-1];

}

}

}

}

int kmp(string s,string sub)

{

pre\_kmp(sub);

int i,j,n,m,cnt;

i=j=cnt=0;

n=s.length();

m=sub.length();

while(i<n){

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

i++;

j++;

}

if(j==m){

cnt++;

j=lps[j-1];

}

else if(i<n&&s[i]!=sub[j]){

if(j==0) i++;

else j=lps[j-1];

}

}

return cnt;

}

int main()

{

int n,i,j,tc,ans;

string s,sub;

cin>>tc;

for(i=1;i<=tc;i++){

cin>>n>>s;

cout<<"Case "<<i<<":\n";

while(n--){

cin>>sub;

ans=kmp(s,sub);

cout<<ans<<endl;

}

}

return 0;

}

**Matrix Exponentiation**

Statement:

f(n)=f(n-1)+f(n-2) ,f(0)=a,f(1)=b,find f(n) where n<=109,

Solution:

const int mxn=1e5+9;

const int eps=1e-9;

ll res[2][2],mod;

void matrix\_mul(ll a[2][2],ll b[2][2],ll sol[2][2])

{

ll i,j,k;

for(i=0;i<2;i++){

for(j=0;j<2;j++){

sol[i][j]=0;

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

sol[i][j]=(sol[i][j]%mod+(a[i][k]\*b[k][j])%mod)%mod;

}

}

}

}

void matrix\_exp(ll a[2][2],ll n)

{

ll i,j,tmp[2][2];

res[0][0]=1;

res[0][1]=0;

res[1][0]=0;

res[1][1]=1;

while(n){

if(n&1){

matrix\_mul(res,a,tmp);

for(i=0;i<2;i++) for(j=0;j<2;j++) res[i][j]=tmp[i][j];

}

matrix\_mul(a,a,tmp);

for(i=0;i<2;i++) for(j=0;j<2;j++) a[i][j]=tmp[i][j];

n>>=1;

}

}

int main()

{

ll n,i,j,t,f0,f1,m,ans,cs=0,a[2][2];

sfl(t);

while(t--){

sfffl(f0,f1,n);sfl(m);

mem(res,0);

mod=1;

for(i=1;i<=m;i++) mod\*=10;

a[0][0]=1;

a[0][1]=1;

a[1][0]=1;

a[1][1]=0;

matrix\_exp(a,n-1);

ans=((res[0][0]\*f1)%mod+(res[0][1]\*f0))%mod;

pf("Case %lld: %lld\n",++cs,ans);

}

return 0;

}

**Digit Dp**

Statement: Given n,Find ,Where f(x)=count of zeroes in x.

Solution:

const int mod=1e9+7;

const int mxn=1e5+9;

const int eps=1e-9;

ll dp[20][20][2][2];

vector<ll> d;

ll digitdp(ll idx,ll check,ll flag,ll sum)

{

ll ans=0,lim,i,next;

if(idx==-1) return sum;

if(dp[idx][sum][check][flag]!=-1) return dp[idx][sum][check][flag];

if(check==0) lim=9;

else lim=d[idx];

for(i=0;i<=lim;i++){

if(i==d[idx]) next=check;

else next=0;

ans+=digitdp(idx-1,next,flag||i!=0,flag?sum+(i==0):0);

}

return dp[idx][sum][check][flag]=ans;

}

int main()

{

ll n,i,j,k,t,ans1,ans2,cs=0;

cin>>t;

while(t--){

cin>>n>>k;

\_case;

n--;

if(n==-1) ans1=-1;

else{

while(n){

d.pb(n%10);

n/=10;

}

mem(dp,-1);

cout<<d.size()-1<<nl;

ans1=digitdp(d.size()-1,1,0,0);

d.clear();

}

while(k){

d.pb(k%10);

k/=10;

}

mem(dp,-1);

ans2=digitdp(d.size()-1,1,0,0);

cout<<ans2-ans1<<nl;

d.clear();

}

return 0;

}