网络流

#include <bits/stdc++.h>

typedef long long ll;

using namespace std;

const ll mn=1e6;

const ll inf=1e18;

ll n,m;

struct ain{

ll v,w,rev;

};

vector<ain>nod[mn];

ll dep[mn];

ll s,en;

ll vis[mn];

void add(ll u,ll v,ll w)

{

nod[u].push\_back({v,1,nod[v].size()});

nod[v].push\_back({u,0,nod[u].size()-1});

}

bool bfs()

{

memset(dep,0,sizeof(dep));

dep[s]=1;

queue<ll>q;

q.push(s);

while(q.size())

{

ll u=q.front();q.pop();

for(auto ed:nod[u])

{

ll v=ed.v,w=ed.w;

if(!dep[v]&&w)

{

dep[v]=dep[u]+1;

q.push(v);

}

}

}

return dep[en]>0;

}

ll dfs(ll u,ll mf)

{

if(u==en) return mf;

ll sum=0;

for(auto &ed:nod[u])

{

ll v=ed.v,w=ed.w,rev=ed.rev;

if(dep[v]==dep[u]+1&&w)

{

ll f=dfs(v,min(mf,w));

ed.w-=f;

nod[v][rev].w+=f;

mf-=f;

sum+=f;

if(mf==0) break;

}

}

if(sum==0) dep[u]=0;

return sum;

}

void dfs1(ll u)

{

vis[u]=1;

for(auto ed:nod[u])

{

ll v=ed.v,w=ed.w,rev=ed.rev;

if(w&&!vis[v])

{

dfs1(v);

}

}

}

ll dinic()

{

ll flow=0;

while(bfs())

{

flow+=dfs(s,inf);

}

return flow;

}

ll u[mn],v[mn];

void solve()

{

cin>>n>>m;

s=1,en=n;

for(ll i=1;i<=m;i++)

{

cin>>u[i]>>v[i];

add(u[i],v[i],1);

}

ll ans=dinic();

cout<<ans<<'\n';

dfs1(s);

for(ll i=1;i<=n;i++)

{

for(auto ed:nod[i])

{

ll v=ed.v,w=ed.w;

if(vis[i]==1&&vis[v]==0)

{

cout<<i<<" "<<v<<'\n';

}

}

}

}

int main()

{

ll t;

t=1;

while(t--)

{

solve();

}

}

St表

#include <bits/stdc++.h>

#include <unordered\_map>

using namespace std;

#define \_\_builtin\_popcount \_\_popcnt

typedef long long ll;

const ll mn = 2e5 + 10;

const ll inf = 1e7;

const ll mod = 1e9 + 7;

ll n,l,r, sum,m, t, a[mn];

ll dp[mn][22];

ll gcd(ll a, ll b)

{

return b == 0 ? a : gcd(b, a % b);

}

void solve()

{

cin >> n >> m;

for (ll i = 1; i <= n; i++)

{

cin >> a[i];

}

for (ll i = 1; i <= n; i++) dp[i][0] = a[i];

for (ll j = 1; j <= log2(n); j++)

{

for (ll i = 1; i + (1 <<j)-1 <= n; i++)

{

dp[i][j] = gcd(dp[i][j - 1], dp[i+(1<<(j-1))][j - 1]);

}

}

for (ll i = 1; i <= m; i++)

{

cin >> l >> r;

ll len = (ll)log2(r - l + 1);

cout << gcd(dp[l][len],dp[r-(1<<len)+1][len] ) << '\n';

}

}

signed main()

{

ios::sync\_with\_stdio(false); cin.tie(0); cout.tie(0);

cin.tie(0);

t = 1;

while (t--)

{

solve();

}

}

Lucas定理

#include <iostream>

using namespace std;

typedef long long ll;

const ll mn=1e5+10;

ll n,m,mod,t,f[mn],g[mn];

ll qpow(ll a,ll b)

{

ll sum=1;

while(b)

{

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

a=a\*a%mod;

b>>=1;

}

return sum%mod;

}

void init()

{

f[0]=g[0]=1;

for(ll i=1;i<=mod;i++)

{

f[i]=f[i-1]\*i%mod;

g[i]=g[i-1]\*qpow(i,mod-2)%mod;

}

}

ll get(ll n,ll m)

{

if(m>n) return 0;

return f[n]\*g[n-m]\*g[m]%mod;

}

ll lucas(ll n,ll m)

{

if(m==0) return 1;

return lucas(n/mod,m/mod)\*get(n%mod,m%mod)%mod;

}

signed main()

{

cin>>t;

while(t--)

{

cin>>n>>m>>mod;

init();

cout<<lucas(n,m)%mod<<'\n';

}

}

矩阵快速幂

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mn = 2e5 + 10;

const ll mod = 1000000007;

ll n,q, f[mn],ma=0;

struct matrix {

ll c[4][4];

matrix() { memset(c, 0, sizeof(c)); }

};

matrix operator\*(matrix &a, matrix &b)

{

matrix t;

for (ll i = 1; i <= 3; i++)

{

for (ll j = 1; j <= 3; j++)

{

for (ll k = 1; k <= 3; k++)

{

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

}

}

}

return t;

}

ll qpow(ll n)//n-3

{

matrix res, base;

res.c[1][1] = res.c[1][2] = res.c[1][3] = 1;

base.c[1][2] = base.c[2][1] = base.c[2][3] = 1; base.c[1][1] = 2; base.c[3][1] = mod-1;

while (n)

{

if (n & 1) res = res \* base;

base=base\*base;

n >>= 1;

}

return res.c[1][1];

}

signed main()

{

ios::sync\_with\_stdio(false); cin.tie(0); cout.tie(0);

cin >> n;

for (ll i = 1; i <= n; i++)

{

cin >> q;

if (q <= 3) cout << 1 << '\n';

else

{

cout<<qpow(q - 3)<<'\n';

}

}

}

卡特兰数

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mn = 1e5 + 10;

const ll inf = 1e18;

const ll mod=1e9+7;

ll t, n, m;

ll a, b,h[mn],g[mn];

ll qpow(ll a,ll b)

{

ll sum=1;

while(b)

{

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

a=a\*a%mod;

b>>=1;

}

return sum;

}

void solve()

{

cin>>n;

cout<<h[n]<<'\n';

}

signed main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

cin>>t;

h[0]=1;

for(ll i=1; i<=mn; i++) h[i]=(h[i-1]%mod\*(4\*i-2)%mod\*qpow(i+1,mod-2))%mod;

while (t--)

{

solve();

}

}

最短路

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

typedef pair<ll, ll> pii;

const ll mn = 2e5 + 10;

const ll inf = 1e10;

ll t,n,m,s,a,b,c,d[mn],vis[mn];

struct ain {

ll v, w;

};

vector<ain>nod[mn];

priority\_queue<pair<ll,ll>>q;

void solve()

{

cin >> n >> m >> s;

for (ll i = 1; i <= m; i++)

{

cin >> a >> b >> c;

nod[a].push\_back({b,c});

}

for (ll i = 0; i <= n; i++) d[i] = inf;

d[s] = 0; q.push({ 0,s });

while (q.size())

{

auto t=q.top(); q.pop();

ll u = t.second;

if (vis[u]) continue;

vis[u] = 1;

for (auto ed : nod[u])

{

ll v = ed.v, w = ed.w;

if (d[v]>d[u]+w)

{

d[v]=d[u]+w;

q.push({ -d[v],v});

}

}

}

for (ll i = 1; i <= n; i++) cout << d[i]<<" ";

}

signed main()

{

t = 1;

while (t--)

{

solve();

}

}

最小生成树

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll mn = 1e6;

const ll inf = 1e10;

ll t, n, k, ans,m,a,b,c,fat[mn];

struct ain {

ll u,v, w;

bool operator <(const ain& t) const {

return w < t.w;

}

}e[mn];

ll find(ll u)

{

if (u == fat[u]) return u;

fat[u] = find(fat[u]);

}

bool kruskal()

{

ll cnt = 0;

sort(e + 1, e + 1 + m);

for (ll i = 1; i <= n; i++) fat[i] = i;

for (ll i = 1; i <= m; i++)

{

ll x = find(e[i].u);

ll y = find(e[i].v);

if (x != y)

{

fat[x] = y;

ans += e[i].w;

cnt++;

}

}

return cnt == n - 1;

}

void solve()

{

cin >> n >> m;

for (ll i = 1; i <= m; i++)

{

cin >> a >> b >> c;

e[i].u = a, e[i].v = b, e[i].w = c;

}

if (kruskal()) cout << ans;

else cout << "orz";

}

signed main()

{

ios::sync\_with\_stdio(false); cin.tie(0); cout.tie(0);

t=1;

while (t--)

{

solve();

}

}

线段树

#include <bits/stdc++.h>

#define lc p<<1

#define rc p<<1|1

#define N 500005

using namespace std;

typedef long long ll;

const ll mn = 1e6+10;

ll k,n, m,a[mn],b[mn],cho,aa,bb;

struct ain {

ll l, r, sum, adt;

}nod[N\*4];

void pushup(ll p)

{

nod[p].sum = nod[lc].sum+ nod[rc].sum;

}

void pushdown(ll p)

{

if (nod[p].adt)

{

nod[lc].sum += (nod[lc].r - nod[lc].l + 1) \* nod[p].adt;

nod[rc].sum += (nod[rc].r - nod[rc].l + 1) \* nod[p].adt;

nod[lc].adt += nod[p].adt;

nod[rc].adt += nod[p].adt;

nod[p].adt = 0;

}

}

void adtd(ll p, ll l, ll r,ll ad)

{

if (l <= nod[p].l && nod[p].r<=r)

{

nod[p].sum += (nod[p].r-nod[p].l+1)\*ad;

nod[p].adt += ad;

return;

}

ll mid = (nod[p].l + nod[p].r) >> 1;

pushdown(p);

if(l<=mid) adtd(lc, l, r,ad);

if(r>mid) adtd(rc, l, r,ad);

pushup(p);

}

ll search(ll p, ll l, ll r)

{

if (l <= nod[p].l && nod[p].r <= r)

{

return nod[p].sum;

}

ll mid =( nod[p].l + nod[p].r )>> 1;

ll sum = 0;

pushdown(p);

if(l<=mid) sum+=search(lc, l, r);

if(r>mid) sum+=search(rc, l, r);

return sum;

}

void build(ll p, ll l, ll r)

{

nod[p] = { l,r,a[l] };

if (l == r)

{

return ;

}

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

build(lc, l, mid);

build(rc, mid + 1, r);

nod[p].sum = nod[lc].sum + nod[rc].sum;

}

signed main()

{

ios::sync\_with\_stdio(false); cin.tie(0); cout.tie(0);

cin >> n>>m;

for (ll i = 1; i <= n; i++) cin >> a[i];

build(1, 1, n);

for (ll i = 1; i <= m; i++)

{

cin >> cho>>aa>>bb;

if (cho == 1)

{

cin >> k;

adtd(1, aa, bb, k);

}

else

{

cout << search(1, aa, bb) << '\n';

}

}

}

Sg函数

 #include <bits/stdc++.h>

 #include<unordered\_map>

 using namespace std;

 typedef long long ll;

 const ll mn=1e6+10;

 const ll inf=1e8;

 const ll mod=998244353;

 ll sg(ll x,ll l,ll r)

 {

 return x%(l+r)/l;

 }



 void solve()

 {

 ll n,l,r,ans=0;

 cin>>n>>l>>r;

 vector<ll>a(n+2);

 for(ll i=1,x;i<=n;i++) cin>>x,ans^=sg(x,l,r);

 if(ans) cout<<"First";

 else cout<<"Second";

 }

 signed main()

 {

 ios::sync\_with\_stdio(false);

 cin.tie(0);

 ll t;

 t=1;

 while(t--)

 {

 solve();

 }

 }

费用流

#include <bits/stdc++.h>

#include<unordered\_map>

using namespace std;

typedef long long ll;

const ll inf=1e9;

const ll mn=1e4+10;

ll n,k,m,s,en;

unordered\_map<string,ll>mp;

struct ain

{

ll v,w,rev,c;

};

vector<vector<ain>>nod;

void add(ll u,ll v,ll w,ll c=0)

{

nod[u].push\_back({v,w,nod[v].size(),c});

nod[v].push\_back({u,0,nod[u].size()-1,-c});

}

ll vis[mn],d[mn],preu[mn],prenum[mn],mf[mn];

bool spfa()

{

fill(d,d+mn,-inf);

fill(mf,mf+mn,inf);

vis[s]=1,d[s]=0;

queue<ll>q;

q.push(s);

while(q.size())

{

ll u=q.front();

q.pop();

vis[u]=0;

ll cnt=-1;

for(auto ed:nod[u])

{

cnt++;

ll v=ed.v,c=ed.c,w=ed.w;

if(d[v]<d[u]+c&&w)

{

d[v]=d[u]+c;

preu[v]=u;

prenum[v]=cnt;

mf[v]=min(mf[u],w);

if(!vis[v]) q.push(v),vis[v]=1;

}

}

}

return d[en]!=-inf;

}

//ll aug(ll u,ll mf,ll &cost)

//{

// if(u==en) return mf;

// ll sum=0;

// vis[u]=1;

// for(auto &ed:nod[u])

// {

// ll v=ed.v,w=ed.w,c=ed.c,rev=ed.rev;

// if(d[v]!=d[u]+c||vis[v]||!w) continue;

// ll f=aug(v,min(mf,w),cost);

// ed.w-=f;

// nod[v][rev].w+=f;

// sum+=f;

// mf-=f;

// cost+=f\*c;

// if(mf==0) break;

// }

// if(sum==0) d[u]=0;

// vis[u]=0;

// return sum;

//}

unordered\_map<ll,string>has;

ll viss[mn];

ll cost=0,flow=0;

void ek()

{

set<ll>ss1;

set<ll,greater<ll>>ss2;

ll cnt=0;

while(spfa())

{

vector<ll>pr;

for(ll v=en;v!=s;v=preu[v])

{

ain &ed=nod[preu[v]][prenum[v]];

ed.w-=mf[en];

nod[v][ed.rev].w+=mf[en];

}

flow+=mf[en];

cost+=mf[en]\*d[en];

}

}

void dfs1(ll u)

{

viss[u]=1;

cout<<has[u-n]<<'\n';

if(u==en-n) return;

for(auto ed:nod[u])

{

ll v=ed.v,w=ed.w;

if(w) continue;

dfs1(ed.v+n);

break;

}

}

void dfs2(ll u)

{

if(u==en-n) return;

for(auto ed:nod[u])

{

ll v=ed.v,w=ed.w;

if(viss[v+n]||w) continue;

dfs2(ed.v+n);

break;

}

cout<<has[u-n]<<'\n';

}

void solve()

{

ll v;

cin>>n>>v;

nod.resize(n\*4);

string st;

for(ll i=1;i<=n;i++)

{

cin>>st,mp[st]=i,has[i]=st;

if(i==1||i==n) add(i,i+n,2,1);

else add(i,i+n,1,1);

}

ll flag=0;

for(ll i=1;i<=v;i++)

{

string s1,s2;

cin>>s1>>s2;

ll v1=mp[s1],v2=mp[s2];

if(v1>v2) swap(v1,v2);

if(v1==1&&v2==n) flag++;

add(v1+n,v2,1,0);

}

s=1,en=2\*n;

ek();

if(flow!=2)

{

if(!flag) cout<<"No Solution!"<<'\n';

else{

cout<<2<<'\n';

cout<<has[1]<<'\n'<<has[n]<<'\n'<<has[1]<<'\n';

}

}

else

{

cout<<cost-2<<'\n';

dfs1(1+n);

dfs2(1+n);

}

}

int main()

{

ll t;

t=1;

ios::sync\_with\_stdio(false);cin.tie(0);

while(t--)

{

solve();

}

system("pause");

}