**KMP：**

void getnext() {

int i,j=0;

next[1]=0;

for(i=2;i<=len2;i++){

while(j>0 && mod[j+1]!=mod[i]) j=next[j];

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

next[i]=j;

}

}

int KMP(){

int i=1,j=0,k=0;

getnext();

while(i<=len1){

while(j>0 && mod[j+1]!=s[i]) j=next[j];

if(mod[j+1]==s[i]){

j++;

if(j==len2) k++;

}

i++;

}

return k;

}

**Manacher**

#include<iostream>

using namespace std;

char s[100010],ss[200010];

int ans,p[200010];

void manacher()

{ int i,len=0,mx=1,idx;ans=0;bool bo=false;

for (i=0;s[i];i++){

ss[++len]='#';

ss[++len]=s[i];

}

ss[++len]='#';ss[++len]=0;ss[0]='$';

p[1]=1;

for (i=2;i<len;i++)

{ if (mx>i)

p[i]=min(p[2\*idx-i],mx-i);

else p[i]=1;

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

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

if (p[i]+i>mx)

{ mx=p[i]+i;

idx=i;

}

}

}

int main(){

while (gets(s))

{ if (s[0]!='\0'){

manacher();

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

}

}

}

**强连通**

#include<cstdio>

#include<cstring>

#include<iostream>

#define N 1010

#define rep(i,u,v) for (int i=(u);i<=(v);i++)

using namespace std;

int n,len,dfn[N],low[N],belong[N],sta[N],instack[N];

int vis,cnt,top,size[N],ru[N],first[N];

struct node

{ int x,y,next;

}edge[1000010];

void ins(int x,int y)

{ edge[++len].x=x;edge[len].y=y;edge[len].next=first[x];first[x]=len;}

void tarjan(int x)

{ dfn[x]=low[x]=++vis;

instack[x]=1;sta[++top]=x;

for (int k=first[x];k;k=edge[k].next)

{ int y=edge[k].y;

if (!dfn[y])

{ tarjan(y);

if (low[x]>low[y])low[x]=low[y];

}

else if (instack[y]&&low[x]>dfn[y])low[x]=dfn[y];

}

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

{ int u; cnt++;

do{

u=sta[top--];

instack[u]=0;

belong[u]=cnt;

size[cnt]++;

}while (x!=u);

}

}

void printff()

{ int tot=0,ans=0;

rep(i,1,cnt)

if (!ru[i]){

tot++;

if (tot>1){

printf("0\n");

return;

}

ans=i;

}

printf("%d\n",size[ans]);

}

int main()

{ int T,k,x;

scanf("%d",&T);

while (T-->0)

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

len=vis=cnt=top=0;

rep(i,1,n){

first[i]=0;

dfn[i]=low[i]=belong[i]=0;

instack[i]=0;ru[i]=size[i]=0;

}

rep(i,1,n)

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

rep(j,1,k){

scanf("%d",&x);

ins(x,i);

}

}

rep(i,1,n)

if (!dfn[i])tarjan(i);

rep(i,1,len)

if (belong[edge[i].x]!=belong[edge[i].y])

ru[belong[edge[i].y]]++;

printff();

}

**}**

**点双联通：**

#include<cstdio>

#include<cstring>

#include<iostream>

#include<cstdlib>

using namespace std;

#define rep(i,u,v) for (int i=(u);i<=(v);i++)

#define N 1010

int bo[N][N],who[N][N],cnt[N],color[N],sta[N],dfn[N],low[N],bk[N];

int topp,n,tot,times;

void tarjan(int u){

int i,v;

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

sta[topp++]=u;

rep(v,1,n)

if(!bo[u][v]){

if(!dfn[v]){

tarjan(v);

if(low[v]<low[u])low[u]=low[v];

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

who[tot][0]=u;

for(i=1,sta[topp]=-1;sta[topp]!=v;i++)

who[tot][i]=sta[--topp];

if(i>2){

cnt[tot]=i;

tot++;

}

}

}

else if(low[u]>dfn[v])low[u]=dfn[v];

}

}

int paint(int fa,int x,int now,int k){

rep(y,0,cnt[k]-1)

if((!bo[who[k][x]][who[k][y]])&&y!=fa){

if(!color[y]){

color[y]=3-now;

if(paint(x,y,3-now,k))return 1;

}

else if(color[x]==color[y])return 1;

}

return 0;

}

int main(){

freopen("ball.in","r",stdin);

freopen("ball.out","w",stdout);

int m,x,y,ans;

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

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

tot=topp=times=0;

rep(i,1,m){

scanf("%d%d",&x,&y);

bo[x][y]=bo[y][x]=1;

}

rep(i,1,n)bo[i][i]=1;

rep(i,1,n)

if(!dfn[i])tarjan(i);

rep(k,0,tot-1){

rep(i,0,cnt[k]-1)color[i]=0;

color[0]=1;

if(paint(-1,0,1,k))

rep(i,0,cnt[k]-1)

bk[who[k][i]]=1;

}

ans=0;

rep(i,1,n)

if(!bk[i])ans++;

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

return 0;

}

前序后序求二叉树的种数

给你前序遍历的结果和后序遍历的结果，求二叉树的种数？

比如：

AC

CA

2

ACK

CKA

1

Input

ACK

KCA

Output

4

【参考程序】:

var n:longint;

s1,s2:string;

function work(x1,y1,x2,y2:longint):longint;

var i,k:longint;

begin

if x1=y1 then work:=1

else

begin

k:=0;

for i:=x2 to y2-1 do

if s2[i]=s1[x1+1] then

begin

k:=i;

break;

end;

if k=y2-1 then work:=work(x1+1,y1,x2,y2-1)\*2

else work:=work(x1+1,x1+1+(k-x2+1)-1,x2,k)\*work(x1+1+(k-x2+1)-1+1,y1,k+1,y2-1);

end;

end;

begin

//while not eof do

//begin

readln(s1);

readln(s2);

n:=length(s1);

writeln(work(1,n,1,n));

//end;

end.

描述 Description

已知一个树的前序遍历和 中序遍历求后序遍历

输入格式 Input Format

第一行，前序遍历访问的节点的 字符序列

第二行，中序遍历访问节点的字符序列（树的深度不大于7）

输出格式 Output Format

如果输入数据准确，输出一行， 后序遍历访问的字符序列

如果输入数据错误，输出“No Solution”

INPUT:

ABDECFG

DBEAFCG

OUTPUT:

DEBFGCA

【参考程序】:

var s1,s2,ss:ansistring;

n:longint;

bo:boolean;

procedure search(x1,y1,x2,y2:longint);

var k,i:longint;

begin

if not bo then exit;

if x1=y1 then ss:=ss+s1[x1]

else

begin

k:=0;

for i:=x2 to y2 do

if s2[i]=s1[x1] then

begin

k:=i;

break;

end;

if k=0 then

begin

bo:=false;

exit;

end;

if k=x2 then

begin

search(x1+1,y1,x2+1,y2);

ss:=ss+s1[x1];

end

else if k=y2 then

begin

search(x1+1,y1,x2,y2-1);

ss:=ss+s1[x1];

end

else

begin

search(x1+1,x1+1+(k-1-x2+1)-1,x2,k-1);

search(x1+1+(k-1-x2+1)-1+1,y1,k+1,y2);

ss:=ss+s1[x1];

end;

end;

end;

begin

while not eof do

begin

readln(s1);

readln(s2);

n:=length(s1);

bo:=true;

ss:='';

search(1,n,1,n);

if not bo then writeln('No Solution')

else writeln(ss);

end;

end.

中序后序求前序

题目:给你二叉树的中序和后序,求它的前序遍历

INPUT:

BCAD

CBDA

OUTPUT:

ABCD

【参考程序】:

var s1,s2:string;

n:longint;

procedure cal(x1,y1,x2,y2:longint);

var i,k:longint;

begin

if x2=y2 then write(s2[y2])//注:哪个一开始确定父亲就输出它的序列

else

begin

k:=0;

for i:=x1 to y1 do//找中序中父亲的位置

if s1[i]=s2[y2] then

begin

k:=i;

break;

end;

if k=y1 then//没有右孩子,递归左孩子

begin

write(s2[y2]);

cal(x1,y1-1,x2,y2-1);

end

else if k=x1 then//没有左孩子,递归右孩子

begin

write(s2[y2]);

cal(k+1,y1,x2,y2-1);

end

else

begin//分别递归左右孩子

write(s2[y2]);

cal(x1,k-1,x2,x2+(k-1-x1+1)-1);

cal(k+1,y1,x2+(k-1-x1+1)-1+1,y2-1);

end;

end;

end;

begin

while not eof do

begin

readln(s1);

readln(s2);

n:=length(s1);

cal(1,n,1,n);

writeln;

end;

end.

**LCA**

#include<iostream>

#include<cmath>

using namespace std;

struct node{int y,d,next;}edge[20010];

struct nnode{int dis,par[18],dep;}t[10010];

int n,m,len,first[10010],maxx;

inline void ins(int x,int y,int d)

{ edge[++len].y=y;edge[len].d=d;

edge[len].next=first[x];first[x]=len;

}

inline void build(int x,int fa,int dis)

{ int i,k;

t[x].par[0]=fa;t[x].dep=t[fa].dep+1;

t[x].dis=dis;maxx=maxx>t[x].dep?maxx:t[x].dep;

for (i=1;t[x].dep-(1<<i)>=1;i++)

t[x].par[i]=t[t[x].par[i-1]].par[i-1];

for (k=first[x];k!=-1;k=edge[k].next)

if (edge[k].y!=fa)build(edge[k].y,x,dis+edge[k].d);

}

inline int LCA(int x,int y)

{ int i;

if (t[x].dep<t[y].dep){int tt;tt=x;x=y;y=tt;}

for(i=maxx;i>=0;i--)

if (t[x].dep-t[y].dep>=(1<<i))

x=t[x].par[i];

if (x==y)return x;

for (i=maxx;i>=0;i--)

if (t[x].dep>(1<<i)&&t[x].par[i]!=t[y].par[i])

{ x=t[x].par[i];y=t[y].par[i];

}

return t[x].par[0];

}

int main()

{ int i,x,y,d,root;len=0;maxx=18;

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

memset(first,-1,(n+1)\*sizeof(int));

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

scanf("%d%d%d",&x,&y,&d);

ins(x,y,d);ins(y,x,d);

}

build(1,0,0);

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

{ scanf("%d%d",&x,&y);

printf("%d\n",t[x].dis+t[y].dis-2\*t[LCA(x,y)].dis);

}

}

**网络流**

#include<iostream>

#include<cstdio>

#include<cstdlib>

using namespace std;

#define rep(i,u,v) for (int i=(u);i<=(v);i++)

#define reph(k,x) for (int k=first[x];k;k=e[k].next)

#define repw(k,x) for (int &k=work[x];k;k=e[k].next)

#define sqr(x) ((x)\*(x))

const long long oo=(long long )1000000000;

struct node{

int y,oth,next;

long long c;

}e[110000];

struct nn{

long long x,y,z;

}a[52];

int n,m,len,st,ed;

long long r;

int first[11000],h[11000],work[11000],list[11000];

long long ans;

void ins(int x,int y,long long c){

int k1=++len,k2=++len;

e[k1].y=y;e[k1].c=c;e[k1].next=first[x];first[x]=k1;

e[k2].y=x;e[k2].c=0;e[k2].next=first[y];first[y]=k2;

e[k1].oth=k2;e[k2].oth=k1;

}

int bfs(){

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

int l=1,r=1;

h[st]=0;list[1]=0;

while (l<=r){

int x=list[l++];

reph(k,x){

int y=e[k].y;

if (h[y]==-1&&e[k].c){

h[y]=h[x]+1;

list[++r]=y;

}

}

}

return h[ed]!=-1;

}

long long dfs(int x,long long flow){

if (x==ed)return flow;

long long ff=0,minn;

repw(k,x){

int y=e[k].y;long long tt=min(e[k].c,flow);

if (h[y]==h[x]+1&&e[k].c)

if (minn=dfs(y,tt)){

e[k].c-=minn;e[e[k].oth].c+=minn;

ff+=minn;flow-=minn;

if (!flow)break;

}

}

return ff;

}

int main(){

int T,y,x;long long tt;

freopen("starwar.in","r",stdin);

freopen("starwar.out","w",stdout);

scanf("%d%d%I64d",&n,&m,&r);

st=0;ed=n+1;r\*=r;

rep(i,1,n){

scanf("%I64d%I64d%I64d",&a[i].x,&a[i].y,&a[i].z);

if (sqr(a[i].x)+sqr(a[i].y)+sqr(a[i].z)>r)ins(i,ed,oo);

}

rep(i,1,m){

scanf("%d%d",&x,&y);

long long t=sqr(a[x].x-a[y].x)+sqr(a[x].y-a[y].y)+sqr(a[x].z-a[y].z);

ins(x,y,t);ins(y,x,t);

}ans=0;

while (bfs()){

memcpy(work,first,sizeof(first));

while(tt=dfs(st,oo))ans+=tt;

}

cout<<ans<<endl;

}

**Km**

/\*

n个人对n个物品，各有一个满意度，分配物品，求最大满意度和

二分图最大权匹配

\*/

#include<iostream>

#include<cstdio>

#include<cstring>

#include<cstdlib>

using namespace std;

#define rep(i,u,v) for (int i=(u);i<=(v);i++)

#define N 102

#define INF 1000000000;

int lx[N],ly[N],vx[N],vy[N];

int match[N],slack[N],weight[N][N];

int n;

int findpath(int x){

vx[x]=1;

rep(y,1,n){

if (!vy[y]){

int temp=lx[x]+ly[y]-weight[x][y];

if (!temp){

vy[y]=1;

if (match[y]==-1||findpath(match[y])){

match[y]=x;

return 1;

}

}

else slack[y]=slack[y]>temp?temp:slack[y];

}

}

return 0;

}

int main(){

while (cin>>n,n){

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

memset(lx,0,sizeof(lx));

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

rep(i,1,n)

rep(j,1,n){

cin>>weight[i][j];

if (lx[i]<weight[i][j])

lx[i]=weight[i][j];

}

rep(i,1,n){

rep(j,1,n)slack[j]=INF;

while (1){

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

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

if (findpath(i))break;//如果找到 匹配，为下一个点找匹配

//否则修改标杆值使有新的边加入二分子图

int inc=INF;

rep(j,1,n){

if (!vy[j]&&slack[j]<inc)

inc=slack[j];

}

rep(j,1,n){

if (vx[j])lx[j]-=inc;

if (vy[j])ly[j]+=inc;

}

}

}

int ans=0;

rep(j,1,n)

if (match[j]!=-1)

ans+=weight[match[j]][j];

cout<<ans<<endl;

}

return 0;

}

**Splay**

#include<cstdio>

#include<cstring>

#include<iostream>

using namespace std;

int root,total,cnt,n,x;

struct node

{ int f,c,son[2],same,dd,sum,maxx,maxl,maxr,rev;

}t[500001];

int que[500001],queue,num[500001];

inline void updatasize(int x)

{ if (x==0)return;

t[x].c=t[t[x].son[1]].c+t[t[x].son[0]].c+1;

t[x].sum=t[x].dd+t[t[x].son[0]].sum+t[t[x].son[1]].sum;

t[x].maxl=max(t[t[x].son[0]].maxl,t[t[x].son[0]].sum+t[x].dd+max(0,t[t[x].son[1]].maxl));

t[x].maxr=max(t[t[x].son[1]].maxr,t[t[x].son[1]].sum+t[x].dd+max(0,t[t[x].son[0]].maxr));

t[x].maxx=max(max(t[t[x].son[0]].maxx,t[t[x].son[1]].maxx),

max(0,t[t[x].son[0]].maxr)+max(0,t[t[x].son[1]].maxl)+t[x].dd);

}

inline void reverse(int x) {

if (x == 0) return;

int tt=t[x].son[1];t[x].son[1]=t[x].son[0];t[x].son[0]=tt;

tt=t[x].maxl;t[x].maxl=t[x].maxr;t[x].maxr=tt;

t[x].rev=1-t[x].rev;

}

inline void makesame(int x,int cc)

{ if (x==0)return;

t[x].dd=cc;

t[x].sum=t[x].c\*t[x].dd;

t[x].maxl=t[x].maxr=t[x].maxx=max(t[x].sum,t[x].dd);

t[x].same=1;

}

inline void push\_down(int x) {

if (x == 0) return;

if (t[x].rev) {

reverse(t[x].son[0]);

reverse(t[x].son[1]);

t[x].rev=0;

}

if(t[x].same)

{ makesame(t[x].son[0],t[x].dd);

makesame(t[x].son[1],t[x].dd);

t[x].same=0;

}

}

inline void rotate(int x,int w)

{

int y=t[x].f;

push\_down(y);push\_down(x);

t[y].son[1-w]=t[x].son[w];

if (t[x].son[w]!=0)t[t[x].son[w]].f=y;

if (t[y].f!=0)

if (t[t[y].f].son[0]==y)t[t[y].f].son[0]=x;

else t[t[y].f].son[1]=x;

t[x].f=t[y].f;

t[x].son[w]=y;

t[y].f=x;

updatasize(y);updatasize(x);

}

inline void splay(int x,int y)

{ push\_down(x);

while(t[x].f!=y)

{ if (t[t[x].f].f==y)

{if (t[t[x].f].son[0]==x)rotate(x,1);else rotate(x,0);}

else {

if (t[t[t[x].f].f].son[0]==t[x].f){

if (t[t[x].f].son[0]==x){rotate(t[x].f,1);rotate(x,1);}

else {rotate(x,0);rotate(x,1);}

}

else {

if (t[t[x].f].son[1]==x){rotate(t[x].f,0);rotate(x,0);}

else {rotate(x,1);rotate(x,0);}

}

}

}

if(y==0)root=x;

}

inline void addnode(int &x,int dd,int y)

{

if (queue!=0){x=que[queue];queue--;}

else x=++total;t[x].rev=0;

t[x].f=y;t[x].dd=dd;

t[x].c=1;t[x].same=0;

t[x].son[0]=t[x].son[1]=0;

t[x].maxx=t[x].maxl=t[x].maxr=dd=t[x].sum;

}

inline void ins(int &x,int l,int r,int y)

{ if (l>r)return;

int mid=(l+r)/2;

addnode(x,num[mid],y);

ins(t[x].son[0],l,mid-1,x);

ins(t[x].son[1],mid+1,r,x);

updatasize(x);

}

inline void select(int k, int f) {

int x = root; k += 1;

while (1) {

push\_down(x);

if (k==t[t[x].son[0]].c + 1) break;

if (k<=t[t[x].son[0]].c) x=t[x].son[0];

else {k-=t[t[x].son[0]].c+1;x=t[x].son[1];}

}

splay(x, f);

}

inline void del(int x)

{ que[++queue]=x;

if (t[x].son[0]!=0)del(t[x].son[0]);

if (t[x].son[1]!=0)del(t[x].son[1]);

}

int main()

{ int i,j,m,x,y,cc;char s[10];

// freopen("sequence.in","r",stdin);

// freopen("sequence.out","w",stdout);

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

t[0].maxl=t[0].maxr=t[0].maxx=-1001;

root=total=queue=0;addnode(root,0,0);

addnode(t[root].son[1],0,root);t[root].c=2;

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

ins(t[2].son[0],1,n,2); updatasize(t[root].son[1]);updatasize(root);

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

{

scanf("%s",s);

if (s[0]=='G'){

scanf("%d%d",&x,&y);

select(x-1,0);

select(x+y,root);

printf("%d\n",t[t[t[root].son[1]].son[0]].sum);

}

else if (s[0]=='M'&&s[2]=='X'){

select(0,0);select(t[root].c-1,root);

printf("%d\n",t[t[t[root].son[1]].son[0]].maxx);

}

else if (s[0]=='I'){

scanf("%d%d",&x,&y);

for (int j=1;j<=y;j++)scanf("%d",&num[j]);

select(x,0);select(x+1,root);

ins(t[t[root].son[1]].son[0],1,y,t[root].son[1]);

updatasize(t[root].son[1]);updatasize(root);

}

else if (s[0]=='D'){

scanf("%d%d",&x,&y);

select(x-1, 0);

select(x+y,root);

del(t[t[root].son[1]].son[0]);t[t[root].son[1]].son[0]=0;

updatasize(t[root].son[1]);updatasize(root);

}

else if (s[0]=='R'){

scanf("%d%d", &x, &y);

select(x-1, 0);

select(x+y,root);

reverse(t[t[root].son[1]].son[0]);

}

else {

scanf("%d%d%d",&x,&y,&cc);

select(x-1,0);

select(x+y,root);

makesame(t[t[root].son[1]].son[0],cc);

}

}

}