回文自动机

```
#include<bits/stdc++.h>
using namespace std;const int N=3e5+7;
char s[N];int i,sz,k,p,now,n,size[N],fa[N],len[N],a[N][26];long long ans;
void add(int c,int n){
	while(s[n-len[p]-1]!=s[n])p=fa[p];//失配后找一个尽量最长的
	if(!a[p][c]){//如果这个回文串没有出现过，说明出现了一个新的本质不同的回文串
		now=++sz;k=fa[p];//新建节点
		len[now]=len[p]+2;
		while(s[n-len[k]-1]!=s[n])k=fa[k];//构建失败指针
		fa[now]=a[k][c];a[p][c]=now;
	}
	p=a[p][c];size[p]++;
}
int main(){
	sz=1;fa[0]=fa[1]=1;len[1]=-1;//开头放一个字符集中没有的字符，减少特判
	scanf("%s",s+1);n=strlen(s+1);
	for(i=1;i<=n;i++)add(s[i]-'a',i);
	for(i=sz;i;i--)size[fa[i]]+=size[i],ans=max(ans,(long long)size[i]*len[i]);
	printf("%lld",ans);
}
```

后缀自动机

```
#include<bits/stdc++.h>
using namespace std;const int N=5e5+7;
int a[N][26],n,m,l[N],r[N],fa[N],b[N],t[N],ans[N],i,p,q,np,nq,last=1,sz=1;char s[N];
void add(int c){
	p=last;np=last=++sz;l[np]=l[p]+1;r[np]=1;
	while(p&&!a[p][c])a[p][c]=np,p=fa[p];
	if(!p)fa[np]=1;
	else{
		q=a[p][c];
		if(l[p]+1==l[q])fa[np]=q;
		else{
			nq=++sz;l[nq]=l[p]+1;
			memcpy(a[nq],a[q],sizeof(a[q]));
			fa[nq]=fa[q];
			fa[np]=fa[q]=nq;
			while(a[p][c]==q)a[p][c]=nq,p=fa[p];
		}
	}
}
int main(){
	for(scanf("%s",s+1),n=strlen(s+1),i=1;i<=n;++i)add(s[i]-'a');
	for(i=1;i<=sz;++i)b[l[i]]++;
	for(i=1;i<=n;++i)b[i]+=b[i-1];
	for(i=1;i<=sz;++i)t[b[l[i]]--]=i;
	for(i=sz;i;--i)r[fa[t[i]]]+=r[t[i]];
	for(i=1;i<=sz;++i)ans[l[i]]=max(ans[l[i]],r[i]);
	for(i=n;i;--i)ans[i]=max(ans[i],ans[i+1]);
	for(i=1;i<=n;++i)printf("%d\n",ans[i]);
}
```

后缀自动机加 **lct**

```
#include<bits/stdc++.h>
using namespace std;const int N=12*1e5;
struct LCT{
	int fa[N],c[N][2],w[N],tag[N],top,q[N];
	bool isroot(int x){return c[fa[x]][0]!=x&&c[fa[x]][1]!=x;}
	void add(int x,int val){if(!x)return;tag[x]+=val;w[x]+=val;}
	void pushdown(int x){if(tag[x])add(c[x][0],tag[x]),add(c[x][1],tag[x]),tag[x]=0;}
```

```
	void rotate(int x){
		int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
		if(!isroot(y))c[z][c[z][1]==y]=x;
		fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
		c[y][l]=c[x][r];c[x][r]=y;
	}
	void splay(int x){
		q[++top]=x;for(int i=x;!isroot(i);i=fa[i])q[++top]=fa[i];
		while(top)pushdown(q[top--]);
		while(!isroot(x)){
			int y=fa[x],z=fa[y];
			if(!isroot(y)){
				if(c[y][0]==x^c[z][0]==y)rotate(x);else rotate(y);
			}rotate(x);
		}
	}
	void access(int x){for(int t=0;x;t=x,x=fa[x])splay(x),c[x][1]=t;}
	void link(int x,int y){access(y);splay(y);add(y,w[x]);fa[x]=y;}
	void cut(int x){access(x);splay(x);add(c[x][0],-w[x]);fa[c[x][0]]=0;c[x][0]=0;}
}t;int a[N][26],fa[N],l[N],np,nq,p,q,sz=1,last=1,n,m,i,j,mask,ans;char s[N*3],str[10];string chars;
void gets(int mask){
	for (chars=s,j=0;j<chars.length();j++){
		mask=(mask*131+j)%chars.length();
		char t=chars[j];
		chars[j]=chars[mask];
		chars[mask]=t;
	}
}
void add(int c){
	p=last;np=last=++sz;l[np]=l[p]+1;t.w[np]=1;
	while(p&&!a[p][c])a[p][c]=np,p=fa[p];
	if(!p)fa[np]=1,t.link(np,1);
	else{
		q=a[p][c];
		if(l[p]+1==l[q])t.link(np,q),fa[np]=q;
		else{
			nq=++sz;l[nq]=l[p]+1;
			memcpy(a[nq],a[q],sizeof(a[q]));
			fa[nq]=fa[q];t.link(nq,fa[nq]);
			fa[np]=fa[q]=nq;t.cut(q);t.link(np,fa[np]);t.link(q,fa[q]);
			while(a[p][c]==q)a[p][c]=nq,p=fa[p];
		}
	}
}
```

```
int main(){
	for(scanf("%d%s",&m,s+1),n=strlen(s+1),i=1;i<=n;++i)add(s[i]-'A');
	for(;m--;){
		scanf("%s%s",str,s);
		if(str[0]=='A'){
			gets(mask);for(i=0;i<chars.size();++i)add(chars[i]-'A');
		}else{
			gets(mask);for(p=1,ans=i=0;i<chars.size();++i){
				j=chars[i]-'A';p=a[p][j];
				if(p==0){ans=-1;break;}
			}
			if(ans==-1)puts("0");else t.splay(p),printf("%d\n",ans=t.w[p]),mask^=ans;
		}
	}
}
```

广义后缀自动机

```
#include<bits/stdc++.h>
using namespace std;const int N=4e6+7,M=1e5+7;
Struct                                                                    data{intto,next;}e[M<<1];int
cnt=1,head[M],c[M],n,m,i,in[N],x,y,p,q,np,nq,sz=1,l[N],fa[N],a[N][11];long long ans;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void add(int last,int c){
	p=last;np=++sz;l[np]=l[p]+1;
	while(p&&!a[p][c])a[p][c]=np,p=fa[p];
	if(!p)fa[np]=1;
	else{
		q=a[p][c];
		if(l[p]+1==l[q])fa[np]=q;
		else{
			nq=++sz;l[nq]=l[p]+1;
			memcpy(a[nq],a[q],sizeof a[q]);
			fa[nq]=fa[q];
			fa[np]=fa[q]=nq;
			while(a[p][c]==q)a[p][c]=nq,p=fa[p];
		}
	}
}
void dfs(int x,int fa,int last){
	add(last,c[x]);last=np;
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa)dfs(e[i].to,x,last);
```

```
}
int main(){
	for(scanf("%d%*d",&n),i=1;i<=n;++i)scanf("%d",c+i);
	for(i=1;i<n;++i)scanf("%d%d",&x,&y),insert(x,y),in[x]++,in[y]++;
	for(i=1;i<=n;++i)if(in[i]==1)dfs(i,0,1);
	for(i=1;i<=sz;++i)ans+=l[i]-l[fa[i]];printf("%lld\n",ans);
}
```

后缀自动机转后缀树

```
#include<bits/stdc++.h>
using namespace std;const int N=1e6+7;typedef long long ll;
struct data{int to,next;}e[N];int a[N][26],n,m,i,j,sz=1,last=1,head[N],cnt,p,q,np,nq,size[N],l[N],fa[N];char s[N];ll ans;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void add(int c){
	p=last;np=last=++sz;l[np]=l[p]+1;size[np]=1;
	while(p&&!a[p][c])a[p][c]=np,p=fa[p];
	if(!p)fa[np]=1;
	else{
		q=a[p][c];
		if(l[p]+1==l[q])fa[np]=q;
		else{
			nq=++sz;l[nq]=l[p]+1;
			memcpy(a[nq],a[q],sizeof(a[q]));
			fa[nq]=fa[q];
			fa[np]=fa[q]=nq;
			while(a[p][c]==q)a[p][c]=nq,p=fa[p];
		}
	}
}
void dfs(int x){
	for(int i=head[x];i;i=e[i].next)dfs(e[i].to),size[x]+=size[e[i].to];
	l[x]-=l[fa[x]];ans-=(ll)size[x]*(size[x]-1)*l[x];
}
int main(){

for(scanf("%s",s+1),n=strlen(s+1),reverse(s+1,s+n+1),i=1;i<=n;++i)add(s[i]-'a');ans=ll(n-1)*n/2*(n+1);
	for(i=2;i<=sz;++i)ins(fa[i],i);dfs(1);printf("%lld\n",ans);
}
```

广义后缀自动机加后缀树加树上 dsu

```
#include<bits/stdc++.h>
using namespace std;const int N=2e5+7;typedef long long ll;ll ans;
struct data{int to,next;}e[N];int a[N][26],n,m,i,j,sum[N],fa[N],l[N],sz=1,last=1,np,p,q,nq,head[N],cnt;set<int>s[N];string str[N];char ss[N];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void add(int c){
	p=last;np=last=++sz;l[np]=l[p]+1;s[np].insert(i);
	while(p&&!a[p][c])a[p][c]=np,p=fa[p];
	if(!p)fa[np]=1;
	else{
		q=a[p][c];
		if(l[p]+1==l[q])fa[np]=q;
		else{
			nq=++sz;l[nq]=l[p]+1;
			memcpy(a[nq],a[q],sizeof(a[q]));
			fa[nq]=fa[q];
			fa[np]=fa[q]=nq;
			while(a[p][c]==q)a[p][c]=nq,p=fa[p];
		}
	}
}
void merge(int x,int y){
	if(s[x].size()<s[y].size())swap(s[x],s[y]);
	for(set<int>::iterator it=s[y].begin();it!=s[y].end();++it)s[x].insert(*it);s[y].clear();
}
void dfs(int x){for(int i=head[x];i;i=e[i].next)dfs(e[i].to),merge(x,e[i].to);sum[x]=s[x].size();}
int main(){
	for(scanf("%d%d",&n,&m),i=1;i<=n;++i)
		for(last=1,scanf("%s",ss),str[i]=ss,j=0;ss[j];++j)add(ss[j]-'a');
	for(i=2;i<=sz;++i)ins(fa[i],i);dfs(1);
	for(i=1;i<=n;printf("%lld%c",ans,i==n?'\n':' '),++i){
		for(j=ans=0,p=1;j<str[i].size();++j){
			p=a[p][str[i][j]-'a'];
			while(sum[p]<m&&p)p=fa[p];
			if(!p)p=1;
			ans+=l[p];
		}
	}
}
```

后缀数组

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;
int xx[N],yy[N],c[N],n,m,i,j,k,p,a[N],b[N],sa[N],height[N],rnk[N],t,w;
void build_sa(int n,int m){
	int*x=xx,*y=yy;
	for(i=0;i<n;++i)c[x[i]=a[i]]++;
	for(i=1;i<m;++i)c[i]+=c[i-1];
	for(i=n-1;i>=0;--i)sa[--c[x[i]]]=i;
	for(k=1;k<n;k<<=1){
		for(p=0,i=n-k;i<n;++i)y[p++]=i;
		for(i=0;i<n;++i)if(sa[i]>=k)y[p++]=sa[i]-k;memset(c,0,sizeof(c));
		for(i=0;i<n;++i)c[x[i]]++;
		for(i=1;i<m;++i)c[i]+=c[i-1];
		for(i=n-1;i>=0;--i)sa[--c[x[y[i]]]]=y[i];
		swap(x,y);p=2;x[sa[0]]=1;
		for(i=1;i<n;++i)x[sa[i]]=y[sa[i-1]]==y[sa[i]]&&y[sa[i-1]+k]==y[sa[i]+k]?p-1:p++;
		if(p>n)break;m=p;
	}
}
void build_height(int n){
	for(i=0;i<n;++i)rnk[sa[i]]=i;
	for(i=j=k=0,a[n]=-1,i=0;i<n;height[rnk[i++]]=k)
	if(rnk[i])for(k?k--:0,j=sa[rnk[i]-1];a[j+k]==a[i+k];k++);else k=0;
}
int main(){
	for(scanf("%d%d",&n,&m),m--,i=0;i<n;++i)scanf("%d",a+i),b[i]=a[i];
	for(sort(b,b+n),i=0;i<n;++i)a[i]=lower_bound(b,b+n,a[i])-b+1;
	for(build_sa(n,n+1),build_height(n),j=i=0;i<n;++i){
		while(t!=w&&xx[w-1]>=height[i])w--;xx[w]=height[i];yy[w++]=i;
		if(i>=m-1)j=max(j,xx[t]);
		if(i>=m-1&&yy[t]==(i-m+1))t++;
	}printf("%d\n",j);
}
```

后缀数组求，循环节最多且字典序最小的子串

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=2e5+7;
int Log[N],bin[20],n,i,j,k,p,mn[N/2][20],t,id,ans,l,r,len1,len2,res,kase;char s[N];
void rmq(int mn[N][20],int a[N]){
	for(i=0;i<n;++i)mn[i][0]=a[i];
	for(i=1;i<=Log[n];++i)
		for(j=0;j+bin[i]<=n;++j)
			mn[j][i]=min(mn[j][i-1],mn[j+bin[i-1]][i-1]);
}
int Q(int mn[N][20],int l,int r){t=Log[r-l+1];return min(mn[l][t],mn[r-bin[t]+1][t]);}
struct SA{
	int height[N],rnk[N],n,m,i,j,xx[N],yy[N],sa[N],c[N],mn[N/2][20];
	void build_sa(int n,int m,char s[N]){
		int*x=xx,*y=yy;memset(xx,0,sizeof(xx));memset(yy,0,sizeof(yy));memset(c,0,sizeof(c));
		for(i=0;i<n;++i)c[x[i]=s[i]]++;
		for(i=1;i<m;++i)c[i]+=c[i-1];
		for(i=n-1;i>=0;--i)sa[--c[x[i]]]=i;
		for(k=1;k<n;k<<=1){
			for(p=0,i=n-k;i<=n-1;++i)y[p++]=i;
			for(i=0;i<n;++i)if(sa[i]>=k)y[p++]=sa[i]-k;memset(c,0,sizeof(c));
			for(i=0;i<n;++i)c[x[i]]++;
			for(i=1;i<m;++i)c[i]+=c[i-1];
			for(i=n-1;i>=0;--i)sa[--c[x[y[i]]]]=y[i];
			swap(x,y);p=2;x[sa[0]]=1;
			for(i=1;i<n;++i)x[sa[i]]=y[sa[i-1]]==y[sa[i]]&&y[sa[i-1]+k]==y[sa[i]+k]?p-1:p++;
			if(p>n)break;m=p;
		}
	}
	void build_height(int n,char s[N]){
		for(i=0;i<n;++i)rnk[sa[i]]=i;
		for(i=k=0;i<n;height[rnk[i++]]=k)
		if(rnk[i])for(k?k--:0,j=sa[rnk[i]-1];s[j+k]==s[i+k];k++);else k=0;
		rmq(mn,height);
	}
	int query(int l,int r){l=rnk[l];r=rnk[r];if(l>r)swap(l,r);l++;return Q(mn,l,r);}
}c[2];
int main(){
	for(bin[0]=1,i=1;i<20;++i)bin[i]=bin[i-1]<<1;
	for(Log[0]=-1,i=1;i<N;++i)Log[i]=Log[i>>1]+1;
```

```
    for(;scanf("%s",s),strcmp(s,"#");){
        n=strlen(s);c[0].build_sa(n,200,s);c[0].build_height(n,s);
        reverse(s,s+n);c[1].build_sa(n,200,s);c[1].build_height(n,s);
        reverse(s,s+n);rmq(mn,c[0].rnk);
        ans=1;id=0;l=c[0].sa[0];r=n-1;
        for(i=1;i<n/2;++i)for(j=0;j<n&&j+i<n;j+=i)if(s[j]==s[j+i]){
            len1=c[0].query(j,j+i);len2=c[1].query(n-j-1,n-(j+i)-1);
            if((k=((len1+len2-1)/i+1))>ans)ans=k,id=N;res=(len1+len2-1)-(k-1)*i;
            if(k==ans&&(p=Q(mn,j-len2+1,j-len2+1+res))<id)id=p,l=c[0].sa[p],r=l+k*i-1;
        }
        for(printf("Case %d: ",++kase),i=l;i<=r;++i)putchar(s[i]);puts("");
    }
}
```

AC 自动机

```
#include<bits/stdc++.h>
using namespace std;const int N=6e3+7,mod=10007;
int a[N][26],v[N],q[N],t,w,sz=1,n,m,i,j,k,l,nxt[N],now,c,dp[107][N],ans;char s[N];
void add(int&x,int v){x+=v%mod;x=(x%mod+mod)%mod;}
void ins(){
    now=1;for(int i=0;s[i];++i){
        c=s[i]-'A';if(!a[now][c])a[now][c]=++sz;now=a[now][c];
    }v[now]=1;
}
void acmach(){
    for(i=0;i<26;++i)a[0][i]=1;t=0;w=1;q[0]=1;
    while(t!=w){
        now=q[t++];
        for(i=0;i<26;++i)if(a[now][i]){
            nxt[a[now][i]]=a[nxt[now]][i];v[a[now][i]]|=v[nxt[a[now][i]]];q[w++]=a[now][i];
        }else a[now][i]=a[nxt[now]][i];
    }
}
int main(){
    for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%s",s),ins();
    for(i=ans=1;i<=m;++i)ans=ans*26%mod;

for(acmach(),dp[0][1]=1,i=0;i<m;++i)for(j=1;j<=sz;++j)for(k=0;k<26;++k)if(!v[a[j][k]])add(dp[i+1][a
[j][k]],dp[i][j]);
    for(i=1;i<=sz;++i)if(!v[i])add(ans,-dp[m][i]);printf("%d\n",ans);
```

```
}
```

AC 自动机加树状数组加树链合并

```
#include<bits/stdc++.h>
using namespace std;const int N=2e6+7,M=1e5+7;
struct data{int to,next;}e[N];int
n,m,i,j,d[N],a[N][26],top,fa[N][22],nxt[N],q[N],id[M],ind,to[M],t,w,bit[N],L[N],R[N],head[N],cnt,sz
=1,c,now,Q;string s;
void add(int x,int v){while(x<=sz)bit[x]+=v,x+=x&-x;}
int sum(int x,int res=0){while(x)res+=bit[x],x-=x&-x;return res;}
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
bool cmp(int x,int y){return L[x]<L[y];}
void dfs(int x){
	for(int i=1;(1<<i)<=d[x];++i)fa[x][i]=fa[fa[x][i-1]][i-1];
	L[x]=++ind;for(int i=head[x];i;i=e[i].next)d[e[i].to]=d[x]+1,fa[e[i].to][0]=x,dfs(e[i].to);R[x]=ind;
}
int lca(int x,int y){
	if(d[x]<d[y])swap(x,y);
	int t=d[x]-d[y];
	for(int i=21;i>=0;--i)if(t>>i&1)x=fa[x][i];
	for(int i=21;i>=0;--i)if(fa[x][i]!=fa[y][i])x=fa[x][i],y=fa[y][i];
	return x==y?x:fa[x][0];
}
void ins(string&s,int num){
	now=1;for(int i=0;i<(int)s.size();++i){
		c=s[i]-'a';if(!a[now][c])a[now][c]=++sz;now=a[now][c];
	}id[num]=now;
}
void acmach(){
	for(i=0;i<26;++i)a[0][i]=1;t=0;w=1;q[0]=1;
	while(t!=w){
		now=q[t++];
		for(i=0;i<26;++i)if(a[now][i]){
			nxt[a[now][i]]=a[nxt[now]][i];q[w++]=a[now][i];
		}else a[now][i]=a[nxt[now]][i];
	}for(i=1;i<=sz;++i)ins(nxt[i],i);
}
int main(){
	for(cin>>n,i=1;i<=n;++i)cin>>s,ins(s,i);acmach();dfs(1);
	for(cin>>m;m--;){
		cin>>Q;
```

```
            if(Q==1){
                cin>>s;top=0;for(i=0,now=1;i<(int)s.size();++i)q[++top]=now=a[now][s[i]-'a'];

for(sort(q+1,q+top+1,cmp),add(L[q[1]],1),i=2;i<=top;++i)add(L[q[i]],1),add(L[lca(q[i],q[i-1])],-1);
            }else cin>>i,printf("%d\n",sum(R[id[i]])-sum(L[id[i]]-1));
        }
}
```

判断是否存在无限长不出现关键字符串的串

```
#include<bits/stdc++.h>
using namespace std;const int N=3e4+7;
int a[N][2],nxt[N],v[N],sz=1,n,m,b[N],d[N],now=1,q[N],i,t,w,c;char s[N];
void ins(){
    now=1;for(int i=0;s[i];++i){
        c=s[i]-'0';if(!a[now][c])a[now][c]=++sz;now=a[now][c];
    }v[now]=1;
}
void acmach(){
    for(i=0;i<2;++i)a[0][i]=1;t=0;w=1;q[0]=1;
    while(t!=w){
        now=q[t++];
        for(i=0;i<2;++i)if(a[now][i]){
            nxt[a[now][i]]=a[nxt[now]][i];v[a[now][i]]|=v[nxt[a[now][i]]];q[w++]=a[now][i];
        }else a[now][i]=a[nxt[now]][i];
    }
}
bool dfs(int x){
    b[x]=1;for(int i=0,y;i<2;++i){
        y=a[x][i];
        if(b[y])return 1;
        if(v[y]||d[y])continue;
        d[y]=1;if(dfs(y))return true;
    }b[x]=0;return false;
}
int main(){
    for(scanf("%d",&n),i=0;i<n;++i)scanf("%s",s),ins();acmach();
    puts(dfs(1)?"TAK":"NIE");
}
```

KMP 加矩阵快速幂

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
inline int read(){
	int f=1,x=0;char ch=getchar();
	while(!(ch>='0'&&ch<='9')){if(ch=='-')f=-1;ch=getchar();}
	while(ch>='0'&&ch<='9'){x=x*10+(ch-'0');ch=getchar();}
	return x*f;
}
int n,m,mod,nxt[25];char ch[25];
int a[25][25],b[25][25];
void mul(int a[25][25],int b[25][25],int ans[25][25]){
	int tmp[25][25];
	for(int i=0;i<m;i++)
		for(int j=0;j<m;j++){
			tmp[i][j]=0;
			for(int k=0;k<m;k++)
				tmp[i][j]=(tmp[i][j]+a[i][k]*b[k][j])%mod;
		}
	for(int i=0;i<m;i++)
		for(int j=0;j<m;j++)
			ans[i][j]=tmp[i][j];
}
int main(){
	n=read();m=read();mod=read();
	scanf("%s",ch+1);
	for(int i=2,j=0;i<=m;i++){
		while(j>0&&ch[j+1]!=ch[i])j=nxt[j];
		if(ch[j+1]==ch[i])j++;
		nxt[i]=j;
	}
	for(int i=0;i<m;i++)
	    for(int j=0;j<=9;j++){
			int t=i;
			while(t>0&&ch[t+1]-'0'!=j)t=nxt[t];
			if(ch[t+1]-'0'==j)t++;
```

```
                if(t!=m)b[t][i]=(b[t][i]+1)%mod;
            }
        for(int i=0;i<m;i++)a[i][i]=1;
        while(n){
            if(n&1)mul(a,b,a);
            mul(b,b,b);
            n>>=1;
        }
        int sum=0;
        for(int i=0;i<m;i++)
            sum=(sum+a[i][0])%mod;
        printf("%d",sum);
        return 0;
}
```

树形 dp 神奇优化

```
#include<bits/stdc++.h>
using namespace std;const int N=1e3+7,inf=-1e9;
int a[N],n,m,i,j,x,s[N][N],dp[N][N],q[N],tot,ans,size[N];
void dfs(int x){q[++tot]=x;size[x]=1;for(int i=1;i<=s[x][0];++i)dfs(s[x][i]),size[x]+=size[s[x][i]];}
int main(){
    for(scanf("%d%d",&n,&m),i=1;i<=n;++i)for(scanf("%d%d",a+i,s[i]),j=1;j<=s[i][0];++j)scanf("%
d",s[i]+j);
    for(i=0;i<=N;++i)for(j=0;j<=m;++j)dp[i][j]=inf;
    for(dfs(1),i=1;i<=n;++i)dp[i][0]=0,dp[i][1]=a[q[i]];
    for(i=n-1;i;--i){
        for(x=q[i],j=1;j<=m;++j)dp[i][j]=max(dp[i][j],max(dp[i+1][j-1]+a[x],dp[i+size[x]][j]));
    }
    for(i=1;i<=m;++i)ans=max(ans,dp[1][i]);
    printf("%d\n",ans);
}
```

spfa 找环

```
#include<cstdio>
#include<cstring>
const int N=57,inf=0x3f3f3f3f,gap=-1e5;
```

```
int
inq[N][N],dis[N][N],flag,xx[N*N],yy[N*N],x,y,t,w,mp[N][N],n,m,i,j,dx[4]={1,-1,0,0},dy[4]={0,0,1,-1};
void spfa(int x,int y){
	if(flag)return;
	inq[x][y]=1;
	for(int i=0;i<4;++i){
		int nx=x+dx[i],ny=y+dy[i];
		if(nx>=1&&nx<=n&&ny>=1&&ny<=m){
			if(dis[nx][ny]>dis[x][y]+mp[nx][ny]){
				if(inq[nx][ny]){
					flag=true;return;
				}else dis[nx][ny]=dis[x][y]+mp[nx][ny],spfa(nx,ny);
			}
		}
	}
	inq[x][y]=0;
}
void spfa(){
	memset(dis,inf,sizeof(dis));memset(inq,0,sizeof(inq));
	t=0;w=1;xx[0]=1;yy[0]=1;inq[1][1]=1;dis[1][1]=mp[1][1];
	while(t!=w){
		x=xx[t];y=yy[t++];if(t==N*N)t=0;inq[x][y]=0;
		for(i=0;i<4;++i){
			int nx=dx[i]+x,ny=dy[i]+y;
			if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&dis[nx][ny]>dis[x][y]+mp[nx][ny]){
				dis[nx][ny]=dis[x][y]+mp[nx][ny];
				if(!inq[nx][ny]){
					inq[nx][ny]=1;
					xx[w]=nx;yy[w]=ny;
					w++;
					if(w==N*N)w=0;
				}
			}
		}
	}
}
int main(){
	for(;~scanf("%d%d",&n,&m);){
		for(i=1;i<=n;++i)for(j=1;j<=m;++j)scanf("%d",&mp[i][j]);

for(memset(dis,0,sizeof(dis)),memset(inq,0,sizeof(inq)),i=1;i<=n&&!flag;++i)for(j=1;j<=m&&!flag;
++j)spfa(i,j);
		if(flag)puts("Doge");else{
			spfa();
```

```
            if(dis[n][m]<gap)puts("Doge");else printf("%d\n",dis[n][m]);
        }
    }
}
```

K 短路

```
#include<bits/stdc++.h>
#define fr first
#define sc second
using namespace std;const int N=1e3+7,M=2e5+7,inf=0x3f3f3f3f;typedef pair<int,int>pa;
struct data{int to,next,v;}e[M];int cnt,d[N],vis[N],n,m,i,j,k,S,T,K;priority_queue<pa,vector<pa>,greater<pa> >q;
struct graph{
    int head[N];
    void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
}g1,g2;
void dijkstra(){
    memset(d,inf,sizeof(d));d[T]=0;q.push(pa(d[T],T));
    while(!q.empty()){
        int x=q.top().sc;q.pop();if(vis[x])continue;vis[x]=1;
        for(int i=g2.head[x];i;i=e[i].next)if(d[e[i].to]>d[x]+e[i].v){
            d[e[i].to]=d[x]+e[i].v;
            q.push(pa(d[e[i].to],e[i].to));
        }
    }
}
int astar(){
    if(d[S]==inf)return -1;q.push(pa(d[S],S));memset(vis,0,sizeof(vis));
    while(!q.empty()){
        int x=q.top().sc,y=q.top().fr;q.pop();
        if(++vis[x]==K&&x==T)return y;
        if(vis[x]<=K)for(int i=g1.head[x];i;i=e[i].next)
            q.push(pa(y-d[x]+e[i].v+d[e[i].to],e[i].to));
    }
    return -1;
}
int main(){
    for(scanf("%d%d",&n,&m);m--;)scanf("%d%d%d",&i,&j,&k),g1.ins(i,j,k),g2.ins(j,i,k);
    scanf("%d%d%d",&S,&T,&K);if(S==T)K++;dijkstra();printf("%d\n",astar());
}
```

分层图最短路

```
#include<bits/stdc++.h>
#define fr first
#define sc second.first
#define tr second.second
using namespace std;const int N=1e5+7,M=N*2;typedef long long ll;const ll inf=0x3f3f3f3f3f3f3f3f;
typedef pair<ll,pair<int,int>>pa;priority_queue<pa,vector<pa>,greater<pa>>q;
struct data{int to,next,v;}e[M<<1];int head[N],i,vis[N][11],cnt,n,m,K,T_T,x,y,z;ll d[N][11],ans;
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void spfa(){
	memset(d,inf,sizeof(d));d[1][0]=0;q.push({0,{1,0}});memset(vis,0,sizeof(vis));
	for(;!q.empty();){
		x=q.top().sc;y=q.top().tr;q.pop();if(vis[x][y])continue;vis[x][y]=1;
		for(i=head[x];i;i=e[i].next){
			if(y!=K&&d[e[i].to][y+1]>d[x][y]){
				d[e[i].to][y+1]=d[x][y];
				q.push({d[e[i].to][y+1],{e[i].to,y+1}});
			}
			if(d[e[i].to][y]>d[x][y]+e[i].v){
				d[e[i].to][y]=d[x][y]+e[i].v;
				q.push({d[e[i].to][y],{e[i].to,y}});
			}
		}
	}
	for(i=0,ans=inf;i<=K;++i)ans=min(ans,d[n][i]);printf("%lld\n",ans);
}
int main(){
	for(scanf("%d",&T_T);T_T--;spfa(),memset(head,0,sizeof(head)),cnt=0){
		for(scanf("%d%d%d",&n,&m,&K),i=1;i<=m;++i)scanf("%d%d%d",&x,&y,&z),ins(x,y,z);
	}
}
```

floyd 求最小环

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=107,inf=0x3f3f3f3f;
int d[N][N],n,m,i,j,k,l,a[N][N],pre[N][N],ans[N],top,res=inf;
```

```
void get(int x,int y){
	if(pre[x][y]==0){ans[++top]=y;return;}
	get(x,pre[x][y]);get(pre[x][y],y);
}
void solve(){
	memset(pre,0,sizeof(pre));res=inf;
	for(scanf("%d",&m),memset(d,inf,sizeof(d)),i=1;i<=n;++i)d[i][i]=0;
	for(i=1;i<=m;++i)scanf("%d%d%d",&j,&k,&l),d[j][k]=d[k][j]=min(d[j][k],l);
	for(memcpy(a,d,sizeof(a)),k=1;k<=n;++k){
		for(i=1;i<k;++i)for(j=i+1;j<k;++j)if((long long)d[i][j]+a[i][k]+a[j][k]<res){
			res=d[i][j]+a[i][k]+a[j][k];top=0;
			ans[++top]=i;
			get(i,j);
			ans[++top]=k;
		}
		for(i=1;i<=n;++i)for(j=1;j<=n;++j)if(d[i][k]+d[k][j]<d[i][j]){
			d[i][j]=d[i][k]+d[k][j];
			pre[i][j]=k;
		}
	}
	if(res==inf)puts("No solution.");
	else for(i=1;i<=top;++i)printf("%d%c",ans[i],i==top?'\n':' ');
}
int main(){for(;scanf("%d",&n),n!=-1;)solve();}
```

floyd 加快速幂

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=2e2+7,M=1e3+7,inf=0x3f3f3f3f;
int H[M],n,m,K,i,j,k,l,s,t,d[N][N],a[N][N];
int id(int x){if(!H[x])H[x]=++n;return H[x]; }
void mul(int A[N][N],int B[N][N],int C[N][N]){
	int tmp[N][N];memset(tmp,inf,sizeof(tmp));
	for(k=1;k<=n;++k)for(i=1;i<=n;++i)for(j=1;j<=n;++j)
	tmp[i][j]=min(tmp[i][j],A[i][k]+B[k][j]);
	for(i=1;i<=n;++i)for(j=1;j<=n;++j)C[i][j]=tmp[i][j];
}
void pow(int A[N][N],int B[N][N]){for(K--;K;K>>=1,mul(A,A,A))if(K&1)mul(B,A,B);}
int main(){
	for(scanf("%d%d%d%d",&K,&m,&s,&t),memset(d,inf,sizeof(d)),i=1;i<=m;++i)
		scanf("%d%d%d",&l,&j,&k),d[id(j)][id(k)]=d[id(k)][id(j)]=l;
```

```
    memcpy(a,d,sizeof(a));pow(d,a);
    printf("%d\n",a[id(s)][id(t)]);
}
```

次小生出树

```
#include<cstdio>
#include<cstring>
#include<algorithm>//mp[i][j]代表 i 与 j 之间的边的长度
using namespace std;const int N=1e2+7;//pre[i] 存 I'
int dp[N][N],mp[N][N],n,m,i,j,T_T,dis[N],pre[N],ans,used[N][N],x,y,z,flag,pos,mn,vis[N];// 如果强行加入 i,j 边，形成环，环上权值的最大值
void Prim(){//used[i][j]代表 i 与 j 之间的边是被最小生成树使用
    memset(dis,0x3f,sizeof(dis));dis[1]=ans=0;memset(pre,0,sizeof(pre));
    memset(dp,0,sizeof(dp));memset(used,0,sizeof(used)); memset(vis,0,sizeof(vis));
    for(i=1;i<=n;++i){
        for(pos=-1,mn=0x3f3f3f3f,x=1;x<=n;++x)if(dis[x]<mn&&!vis[x])pos=i,mn=dis[x];
        ans+=mn;used[pre[pos]][pos]=used[pos][pre[pos]]=1;vis[pos]=1;
        for(x=1;x<=n;++x){
            if(!vis[x]&&mp[x][pos]<dis[x]){
                dis[x]=mp[x][pos];
                pre[x]=pos;
            }
            if(vis[x])dp[x][pos]=dp[pos][x]=max(mn,dp[pre[pos]][x]);
        }
    }
}
int main(){
    for(scanf("%d",&T_T);T_T--;){
        for(memset(mp,0x3f,sizeof(mp)),flag=0,scanf("%d%d",&n,&m),i=1;i<=m;++i)
            scanf("%d%d%d",&x,&y,&z),mp[x][y]=mp[y][x]=min(mp[x][y],z);
        Prim();

    for(i=1;i<=n&&!flag;++i)for(j=i+1;j<=n&&!flag;++j)if(!used[i][j]&&dp[i][j]==mp[i][j])flag=true;
        if(flag)puts("Not Unique!");
        else printf("%d\n",ans);
    }
}
```

kruskal 算法最小生成树

```
#include<bits/stdc++.h>
#define fr first
#define sc second.first
#define tr second.second
using namespace std;const int N=1e4+7;typedef pair<int,pair<int,int> >pa;
int T_T,n,i,j,x,y,z,fa[N],size[N];pa s[N];typedef long long ll;ll ans;
int find(int x){return x==fa[x]?x:fa[x]=find(fa[x]);}
int main(){
	for(scanf("%d",&T_T);T_T--;printf("%lld\n",ans),ans=0){
		for(scanf("%d",&n),i=1;i<=n;++i)fa[i]=i,size[i]=1;
		for(i=1;i<n;++i)scanf("%d%d%d",&s[i].sc,&s[i].tr,&s[i].fr);
		for(sort(s+1,s+n),i=1;i<n;++i){
			x=s[i].sc;y=s[i].tr;z=s[i].fr;x=find(x);y=find(y);
			ans+=ll(z+1)*(size[x]*size[y]-1);
			fa[x]=y;
			size[y]+=size[x];
		}
	}
}
```

k 小限制度生成树

```
#include<bits/stdc++.h>
#define fr first
#define sc second.first
#define tr second.second
using namespace std;const int N=1e2+7,M=1e5+7;typedef pair<int,pair<int,int> >pa;
int	fa[N],i,j,k,dp[N],ans,id,S,T_T,n,G[N][N],used[N][N],m,sum,cnt,K;map<string,int>mp;string
a,b;pa e[M],mx[N];
typedef pair<int,int>node;vector<node>vec[N];
int find(int x){return x==fa[x]?x:fa[x]=find(fa[x]);}
void ins(int u,int v,int w){vec[u].push_back(node(v,w));used[u][v]=1;}
void del(int x,int y){
	used[x][y]=0;
	for(int i=0;i<vec[x].size();++i)if(vec[x][i].first==y){
		vec[x].erase(vec[x].begin()+i);return;
	}
}
void Del(int x,int y){del(x,y);del(y,x);}
```

```
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,w);}
int ID(string a){if(!mp[a])mp[a]=++n;return mp[a];}
void dfs(int x,int fa=0){
	for(int i=0;i<vec[x].size();++i){
		int y=vec[x][i].first,z=vec[x][i].second;
		if(y==fa)continue;
		if(z>dp[x]){
			dp[y]=z;
			mx[y].fr=z;mx[y].sc=x;mx[y].tr=y;
		}else {
			dp[y]=dp[x];
			mx[y]=mx[x];
		}dfs(y,x);
	}
}
void solve(){
	for(cin>>m,sum=n=cnt=0,memset(G,-1,sizeof(G)),memset(used,0,sizeof(used)),mp.clear(),i=
1;i<=m;++i)
	cin>>a>>b>>e[i].fr,e[i].sc=ID(a),e[i].tr=ID(b),G[ID(a)][ID(b)]=G[ID(b)][ID(a)]=e[i].fr;
	for(i=1;i<=n;++i)fa[i]=i,vec[i].clear();
	for(S=ID("Park"),sort(e+1,e+m+1),i=1;i<=m;++i)if(e[i].sc!=S&&e[i].tr!=S){
		j=find(e[i].sc);k=find(e[i].tr);
		if(j!=k)fa[j]=k,sum+=e[i].fr,insert(e[i].sc,e[i].tr,e[i].fr);
	}
	for(i=1;i<=m;++i)if(e[i].sc==S||e[i].tr==S){
		j=find(e[i].sc);k=find(e[i].tr);
		if(j!=k)fa[j]=k,sum+=e[i].fr,cnt++,insert(e[i].sc,e[i].tr,e[i].fr);
	}
	for(cin>>K,cnt++;cnt<=K;++cnt){
		memset(dp,0,sizeof(dp));dfs(S);
		for(i=1,id=ans=0;i<=n;++i)if(G[S][i]!=-1&&!used[S][i]){
			if(dp[i]-G[S][i]>ans){
				ans=dp[i]-G[S][i];
				id=i;
			}
		}if(ans==0)break;
		sum-=ans;
		Del(mx[id].sc,mx[id].tr);
		insert(S,id,G[S][id]);
	}printf("Total miles driven: %d\n",sum);
}
int main(){
	for(scanf("%d",&T_T);T_T--;){
		solve();
```

```
            if(T_T)puts("");
        }
}
```

最优比率生成树

```
#include<cstdio>
#include<cstring>
#include<algorithm>
#define sqr(x) ((x)*(x))
using namespace std;const int N=1e3+7,inf=1e9;
int X[N],Y[N],Z[N],n,i,j,k,vis[N];double dp[N],mn,l,r,mid,res;
double dis(int a,int b){return sqrt(sqr(X[a]-X[b])+sqr(Y[a]-Y[b]));}
bool Prime(double mid){
        memset(vis,0,sizeof(vis));for(i=1;i<=n;++i)dp[i]=inf;dp[1]=res=0;
        for(i=1;i<=n;++i){
                for(j=1,k=0,mn=inf;j<=n;++j)if(dp[j]<mn&&!vis[j])mn=dp[j],k=j;
                vis[k]=1;res+=mn;
                for(j=1;j<=n;++j)if(!vis[j]&&dp[j]>abs(Z[j]-Z[k])-mid*dis(j,k))
                        dp[j]=abs(Z[j]-Z[k])-mid*dis(j,k);
        }
        return res>=0;
}
int main(){
        for(;scanf("%d",&n),n;printf("%.3f\n",l)){
                for(i=1;i<=n;++i)scanf("%d%d%d",X+i,Y+i,Z+i);
                for(l=0,r=1e3;r-l>=1e-4;)if(Prime(mid=(l+r)/2))l=mid;else r=mid;
        }
}
```

最优比率生成树迭代版

```
#include<cstdio>
#include<cmath>
#include<cstring>
using namespace std;
const int N=1e3+7,inf=1e9;
#define sqr(x) ((x)*(x))
double dp[N],d[N],p[N],l,r,cost,length,mn;int n,i,j,k,X[N],Y[N],Z[N],vis[N];
double dis(int a,int b){return sqrt(sqr(X[a]-X[b])+sqr(Y[a]-Y[b]));}
double prim(double mid){
```

```
	for(i=1;i<=n;++i)dp[i]=inf,vis[i]=0;dp[1]=cost=length=0;
	for(i=1;i<=n;++i){
		for(j=1,mn=inf,k=0;j<=n;++j)if(!vis[j]&&dp[j]<mn)k=j,mn=dp[j];
		vis[k]=1;cost+=d[k];length+=p[k];
		for(j=1;j<=n;++j)if(!vis[j]&&dp[j]>fabs(0.0+Z[j]-Z[k])-mid*dis(j,k)){
			dp[j]=fabs(0.0+Z[j]-Z[k])-mid*dis(j,k);
			d[j]=fabs(Z[j]-Z[k]);
			p[j]=dis(j,k);
		}
	}
	return cost/length;
}
int main(){
	for(;scanf("%d",&n),n;printf("%.3f\n",l)){
		for(i=1;i<=n;++i)scanf("%d%d%d",X+i,Y+i,Z+i);
		for(l=r=0;;){
			r=prim(l);if(fabs(r-l)<=1e-4)break;l=r;
		}
	}
}
```

树的直径

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;
struct data{int to,next,v;}e[N<<1];int head[N],cnt=1,mx,n,K,i,j,x,y,dia,tot,s1[N],s2[N];
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,w);}
int dfs(int x,int fa=0){
	int mx1=0,mx2=0;
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa){
		int v=e[i].v+dfs(e[i].to,x);
		if(v>mx1)s2[x]=s1[x],s1[x]=i,mx2=mx1,mx1=v;
		else if(v>mx2)s2[x]=i,mx2=v;
	}
	if(mx1+mx2>dia)dia=mx1+mx2,mx=x;
	return mx1;
}
int main(){
	for(scanf("%d%d",&n,&K),tot=2*n-2,i=1;i<n;++i)scanf("%d%d",&x,&y),insert(x,y,1);
	dfs(1);tot=tot-dia+1;if(K==2){
		dia=0;
		for(i=s1[mx];i;i=s1[e[i].to])e[i].v=e[i^1].v=-1;
```

```
        for(i=s2[mx];i;i=s1[e[i].to])e[i].v=e[i^1].v=-1;
        dfs(1);tot=tot-dia+1;
    }printf("%d\n",tot);
}
```

有上下届，无源汇的网络流

```
#include<bits/stdc++.h>
using namespace std;const int N=207,M=1e5+7,inf=1e9+7;
struct data{int to,next,v;}e[M];int head[N],cnt=1,d[N],flag,S,T,q[N],low[M],t,w,T_T,n,m,i,j,x,y,z1,z2,in[N];
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,0);}
bool bfs(){
    memset(d,-1,sizeof(d));t=0;w=1;q[0]=S;d[S]=0;
    while(t!=w){
        int now=q[t++];
        for(int i=head[now];i;i=e[i].next)if(d[e[i].to]==-1&&e[i].v)
        q[w++]=e[i].to,d[e[i].to]=d[now]+1;
    }
    return d[T]!=-1;
}
int dfs(int x,int f){
    if(x==T)return f;
    int w,used=0;
    for(int i=head[x];i;i=e[i].next)if(e[i].v&&d[e[i].to]==d[x]+1){
        w=dfs(e[i].to,min(f-used,e[i].v));
        used+=w;e[i].v-=w;e[i^1].v+=w;
        if(used==f)return f;
    }
    if(!used)d[x]=-1;
    return used;
}
void dinic(){while(bfs())dfs(S,inf);}
int main(){
    for(scanf("%d",&T_T);T_T--;cnt=1,memset(in,0,sizeof(in)),memset(head,0,sizeof(head))){
        for(scanf("%d%d",&n,&m),S=0,T=n+1,i=1;i<=m;++i){
            scanf("%d%d%d%d",&x,&y,&z1,&z2);low[i]=z1;
            in[x]-=z1;in[y]+=z1;insert(x,y,z2-z1);
        }
        for(i=1;i<=n;++i)if(in[i]<0)insert(i,T,-in[i]);else insert(S,i,in[i]);
        for(dinic(),flag=1,i=head[S];i;i=e[i].next)if(e[i].v)flag=0;
        if(!flag)puts("NO");
```

```
            else{
                puts("YES");
                for(i=1;i<=m;++i)printf("%d\n",e[(i<<1)^1].v+low[i]);
            }
        }
}
```

有源汇，有上下界的网络流

```
#include<bits/stdc++.h>
using namespace std;const int N=207,M=1e5+7,inf=1e9;
struct data{int to,next,v;}e[M];int
head[N],in[N],cnt=1,ans,flag,n,m,i,j,x,y,z1,z2,S,T,U,V,t,w,q[N],d[N];
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,0);}
bool bfs(){
    memset(d,-1,sizeof(d));t=0;w=1;q[0]=S;d[S]=0;
    while(t!=w){
        int now=q[t++];
        for(int i=head[now];i;i=e[i].next)if(d[e[i].to]==-1&&e[i].v)
        q[w++]=e[i].to,d[e[i].to]=d[now]+1;
    }
    return d[T]!=-1;
}
int dfs(int x,int f){
    if(x==T)return f;
    int w,used=0;
    for(int i=head[x];i;i=e[i].next)if(d[e[i].to]==d[x]+1&&e[i].v){
        w=dfs(e[i].to,min(f-used,e[i].v));
        used+=w;e[i].v-=w;e[i^1].v+=w;
        if(used==f)return f;
    }
    if(!used)d[x]=-1;
    return used;
}
int dinic(){int ans=0;while(bfs())ans+=dfs(S,inf);return ans;}
int main(){
    for(scanf("%d%d%d%d",&n,&m,&U,&V),S=0,T=n+1,i=1;i<=m;++i){
        scanf("%d%d%d%d",&x,&y,&z1,&z2);insert(x,y,z2-z1);
        in[x]-=z1;in[y]+=z1;
    }
    for(i=1;i<=n;++i)if(in[i]<0)insert(i,T,-in[i]);else insert(S,i,in[i]),ans+=in[i];
```

```
	insert(V,U,inf);
	if(dinic()!=ans)puts("please go home to sleep");
	else S=U,T=V,printf("%d\n",dinic());
}
```

有源汇有上下界的网络流

```
#include<bits/stdc++.h>
using namespace std;const int N=2e3+7,M=1e6+7,inf=1e9+7;
struct data{int to,next,v;}e[M<<1];int
head[N],n,m,cnt=1,ind=0,q[N],i,j,x,low[M],up,day[N],S,T,SS,TT,t,w,d[N],g[N],ans,in[N];
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,0);}
bool bfs(){
	memset(d,-1,sizeof(d));t=0;w=1;q[0]=S;d[S]=0;
	while(t!=w){
		int now=q[t++];
		for(int i=head[now];i;i=e[i].next)if(d[e[i].to]==-1&&e[i].v)
		q[w++]=e[i].to,d[e[i].to]=d[now]+1;
	}
	return d[T]!=-1;
}
int dfs(int x,int f){
	if(x==T)return f;
	int w,used=0;
	for(int i=head[x];i;i=e[i].next)if(e[i].v&&d[e[i].to]==d[x]+1){
		w=dfs(e[i].to,min(f-used,e[i].v));
		used+=w;e[i].v-=w;e[i^1].v+=w;if(used==f)return f;
	}
	if(!used)d[x]=-1;return used;
}
void dinic(){while(bfs())ans+=dfs(S,inf);}
int main(){

for(;~scanf("%d%d",&n,&m);puts(""),memset(head,0,sizeof(head)),ind=ans=0,cnt=1,memset(in,0,
sizeof(in))){

for(SS=n+m+1,TT=n+m+2,S=0,T=TT+1,i=1;i<=m;++i)scanf("%d",g+i),in[i+n]-=g[i],in[TT]+=g[i];
		for(i=1;i<=n;++i){
			for(scanf("%d%d",&j,day+i);j--;){
				++ind;scanf("%d%d%d",&x,low+ind,&up);++x;
				in[x+n]+=low[ind];in[i]-=low[ind];
```

```
                insert(i,x+n,up-low[ind]);
            }
        }
        for(i=1;i<=n;++i)insert(SS,i,day[i]);for(i=1;i<=m;++i)insert(i+n,TT,inf);
        for(i=1;i<=TT;++i)if(in[i]<0)insert(i,T,-in[i]);else insert(S,i,in[i]),ans-=in[i];
        insert(TT,SS,inf);dinic();if(ans!=0)puts("-1");
        else{
            S=SS;T=TT;dinic();printf("%d\n",ans);
            for(i=1;i<=ind;++i)printf("%d\n",e[(i<<1)^1].v+low[i]);
        }
    }
}
```

bzoj3876 有源汇的有上下界的费用流

```
#include<bits/stdc++.h>
using namespace std;const int N=307,M=2e5+7,inf=0x3f3f3f3f;
struct data{int from,to,next,v,c;}e[M<<1];int
head[N],cnt=1,from[N],d[N],q[N],t,w,n,m,i,j,inq[N],x,y,S,T,in[N],ans;
void ins(int u,int v,int w,int
c){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;e[cnt].c=c;e[cnt].from=u;}
void insert(int u,int v,int w,int c){ins(u,v,w,c);ins(v,u,0,-c);}
bool spfa(){
    memset(d,inf,sizeof(d));t=0;w=1;q[0]=S;inq[S]=1;d[S]=0;
    for(;t!=w;){
        x=q[t++];if(t==N)t=0;inq[x]=0;
        for(i=head[x];i;i=e[i].next)if(e[i].v&&d[e[i].to]>d[x]+e[i].c){
            d[e[i].to]=d[x]+e[i].c;
            from[e[i].to]=i;
            if(!inq[e[i].to]){inq[e[i].to]=1;q[w++]=e[i].to;if(w==N)w=0;}
        }
    }return d[T]!=inf;
}
void mincf(){
    x=inf;
    for(i=from[T];i;i=from[e[i].from])x=min(x,e[i].v);
    for(i=from[T];i;i=from[e[i].from])e[i].v-=x,e[i^1].v+=x,ans+=x*e[i].c;
}
int main(){
    for(scanf("%d",&n),T=n+1,i=1;i<=n;++i){
        for(scanf("%d",&j);j--;){
            scanf("%d%d",&x,&y);
```

```
                in[i]--;in[x]++;insert(i,x,inf,y);ans+=y;
            }
            insert(i,1,inf,0);
        }
        for(i=1;i<=n;++i)if(in[i]<0)insert(i,T,-in[i],0);else insert(S,i,in[i],0);
        while(spfa())mincf();printf("%d\n",ans);
}

#include<bits/stdc++.h>
using namespace std;const int N=400,M=1e5+7,inf=0x3f3f3f3f;
struct data{int to,next,v,c;}e[M];int head[N],cnt=1,ans,S,T,n,m,i,j,k,x,y,t,w,in[N],inq[N],q[N],d[N];
void ins(int u,int v,int w,int c){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;e[cnt].c=c;}
void insert(int u,int v,int w,int c){ins(u,v,w,c);ins(v,u,0,-c);}
bool spfa(){
        memset(d,inf,sizeof(d));memset(inq,0,sizeof(inq));t=0;w=1;q[0]=T;inq[T]=1;d[T]=0;
        while(t!=w){
            x=q[t++];if(t==N)t=0;inq[x]=0;
            for(i=head[x];i;i=e[i].next)if(e[i^1].v&&d[e[i].to]>d[x]-e[i].c){
                d[e[i].to]=d[x]-e[i].c;
                if(!inq[e[i].to]){inq[e[i].to]=1;q[w++]=e[i].to;if(w==N)w=0;}
            }
        }
        return d[S]!=inf;
}
int dfs(int x,int f){
        inq[x]=1;
        if(x==T)return f;
        int w,used=0;
        for(int i=head[x];i;i=e[i].next)if(!inq[e[i].to]&&d[e[i].to]==d[x]-e[i].c&&e[i].v){
            w=dfs(e[i].to,min(f-used,e[i].v));
            used+=w;e[i].v-=w;e[i^1].v+=w;ans+=w*e[i].c;
            if(used==f)return f;
        }
        if(!used)d[x]=inf;
        return used;
}
void zkw(){
        while(spfa()){
            inq[T]=1;
            while(inq[T]){
                memset(inq,0,sizeof(inq));
                dfs(S,inf);
            }
```

```
	}
}
int main(){
	for(scanf("%d",&n),T=n+1,i=1;i<=n;++i){
		for(scanf("%d",&k);k--;){
			scanf("%d%d",&x,&y);
			in[i]--;in[x]++;ans+=y;
			insert(i,x,inf,y);
		}
		if(i!=1)insert(i,1,inf,0);
	}
	for(i=1;i<=n;++i)if(in[i]<0)insert(i,T,-in[i],0);else insert(S,i,in[i],0);
	zkw();printf("%d\n",ans);
}
```

最大权闭包

```
#include<bits/stdc++.h>
using namespace std;const int N=407,M=N*N,inf=1e9;
struct data{int to,next,v;}e[M<<1];int cnt=1,d[N],q[N],head[N],t,w,S,T,i,j,x,ans,n,m;
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,0);}
bool bfs(){
	memset(d,-1,sizeof(d));t=0;w=1;q[0]=S;d[S]=0;
	while(t!=w){
		int now=q[t++];

for(i=head[now];i;i=e[i].next)if(d[e[i].to]==-1&&e[i].v)d[e[i].to]=d[now]+1,q[w++]=e[i].to;
	}return d[T]!=-1;
}
int dfs(int x,int f){
	if(x==T)return f;
	int w,used=0;
	for(int i=head[x];i;i=e[i].next)if(d[e[i].to]==d[x]+1&&e[i].v){
		w=dfs(e[i].to,min(f-used,e[i].v));
		used+=w;e[i].v-=w;e[i^1].v+=w;
		if(used==f)return f;
	}
	if(!used)d[x]=-1;return used;
}
void dinic(){while(bfs())ans-=dfs(S,inf);}
int main(){
	for(scanf("%d%d",&n,&m),T=n+m+1,i=1;i<=m;++i)scanf("%d",&j),insert(n+i,T,j);
```

```
for(i=1;i<=n;++i)for(scanf("%d%d",&x,&j),ans+=x,insert(S,i,x);j--;)scanf("%d",&x),insert(i,x+n,inf);
	dinic();printf("%d\n",ans);
}
```

二分图染色

```
#include<bits/stdc++.h>
using namespace std;const int N=2e4+7,M=2e5+7;
struct data{int to,next;}e[M];int head[N],cnt,n,m,i,j,X[M],Y[M],Z[M],col[N],ans,l,r,mid,flag;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs(int x){
	if(!flag)return;
	for(int i=head[x];i;i=e[i].next)if(!col[e[i].to])col[e[i].to]=-col[x],dfs(e[i].to);
	else if(col[x]!=-col[e[i].to])flag=0;
}
bool check(int x){
	memset(head,0,sizeof(head));cnt=0;flag=1;memset(col,0,sizeof(col));
	for(i=1;i<=m;++i)if(Z[i]>x)insert(X[i],Y[i]);
	for(i=1;i<=n;++i)if(!col[i])col[i]=-1,dfs(i);
	return flag;
}
int main(){
	for(scanf("%d%d",&n,&m),i=1;i<=m;++i)scanf("%d%d%d",X+i,Y+i,Z+i);
	for(l=0,r=1<<30;l<=r;)check(mid=l+r>>1)?(r=(ans=mid)-1):(l=mid+1);
	printf("%d\n",ans);return 0;
}
```

//KM 最大匹配

```
#include<bits/stdc++.h>
using namespace std;const int N=307,inf=0x3f3f3f3f;
int slack[N],va[N],vb[N],delta,la[N],lb[N],lnk[N],n,m,i,j,ans,mp[N][N];
bool dfs(int x){
	va[x]=1;
	for(int i=1;i<=n;++i)if(!vb[i])if(la[x]+lb[i]==mp[x][i]){
		vb[i]=1;
		if(!lnk[i]||dfs(lnk[i])){lnk[i]=x;return true;}
```

```
        }else slack[i]=min(slack[i],la[x]+lb[i]-mp[x][i]);
        return false;
}
void km(){
        for(i=1;i<=n;++i)lb[i]=0,la[i]=*max_element(mp[i]+1,mp[i]+n+1);memset(lnk,0,sizeof(lnk));
        for(i=1;i<=n;++i){
                memset(slack,inf,sizeof(slack));
                while(true){
                        memset(va,0,sizeof(va));memset(vb,0,sizeof(vb));
                        if(dfs(i))break;
                        for(delta=inf,j=1;j<=n;++j)if(!vb[j])delta=min(delta,slack[j]);
                        for(j=1;j<=n;++j){
                                if(va[j])la[j]-=delta;
                                if(vb[j])lb[j]+=delta;else slack[j]-=delta;
                        }
                }
        }
        for(ans=0,i=1;i<=n;++i)ans+=mp[lnk[i]][i];printf("%d\n",ans);
}
int main(){
        for(;~scanf("%d",&n);km())for(i=1;i<=n;++i)for(j=1;j<=n;++j)scanf("%d",&mp[i][j]);
}
```

二分图最小顶点覆盖，选择几个点集 S,使得所有的边都与点集 S 有关。求点数最小的 S

```
#include<cstdio>
#include<cstring>
using namespace std;const int N=2507,M=57;
struct data{int to,next;}e[N*2];int head[N],cnt,vis[N],lnk[N],i,j,k,ida[M][M],idb[M][M],cnta,cntb,n,m,ans;char s[M][M];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
bool dfs(int x){
        for(int i=head[x];i;i=e[i].next)if(!vis[e[i].to]){
                vis[e[i].to]=1;
                if(!lnk[e[i].to]||dfs(lnk[e[i].to])){lnk[e[i].to]=x;return true;}
        }return false;
}
void hungary(){memset(lnk,0,sizeof(lnk));for(int i=1;i<=cnta;memset(vis,0,sizeof(vis)),++i)if(dfs(i))ans++;}
int main(){

for(;~scanf("%d%d",&n,&m);hungary(),printf("%d\n",ans),cnta=cntb=cnt=ans=0,memset(head,0,sizeof(head))){
                for(i=1;i<=n;++i)scanf("%s",s[i]+1);
                for(i=1;i<=n;++i){
                        for(j=1;j<=m;j=k){
```

```
                for(;j<=m&&s[i][j]=='.';++j);
                if(j==m+1)break;
                cnta++;
                for(k=j;s[i][k]=='*';++k)ida[i][k]=cnta;
            }
        }
        for(i=1;i<=m;++i){
            for(j=1;j<=n;j=k){
                for(;j<=n&&s[j][i]=='.';++j);
                if(j==n+1)break;
                cntb++;
                for(k=j;s[k][i]=='*';++k)idb[k][i]=cntb;
            }
        }
        for(i=1;i<=n;++i)for(j=1;j<=m;++j)if(s[i][j]=='*')ins(ida[i][j],idb[i][j]);
    }
}
```

//二分图最大独立集，等于所有顶点数-最大匹配数。补图的最大团，等于最大独立集

```
#include<cstdio>
#include<cstring>
const int N=507;
int mp[N][N],n,m,i,j,k,ans,lnk[N],vis[N];
bool dfs(int x){
    for(int i=1;i<=n;++i)if(!vis[i]&&mp[x][i]){
        vis[i]=1;
        if(!lnk[i]||dfs(lnk[i])){lnk[i]=x;return true;}
    }return false;
}
void
hungary(){memset(lnk,0,sizeof(lnk));for(i=1;i<=n;++i,memset(vis,0,sizeof(vis)))if(dfs(i))ans++;}
int main(){
    for(;~scanf("%d",&n);hungary(),printf("%d\n",n-ans/2),ans=0){
        for(memset(mp,0,sizeof(mp)),i=1;i<=n;++i)
            for(scanf("%*d %*c %*c %d %*c",&j);j--;)scanf("%d",&k),mp[i][k+1]=1;
    }
}
```

有向无环图的最小链覆盖，等于有向无环图的最长反链

```
#include<bits/stdc++.h>
using namespace std;const int N=1e2+7;
int mp[N][N],n,m,i,j,k,vis[N],lnk[N],ans;
bool dfs(int x){
    for(int i=1;i<=n;++i)if(!vis[i]&&mp[x][i]){
        vis[i]=1;
        if(!lnk[i]||dfs(lnk[i])){lnk[i]=x;return true;}
```

```
	}return false;
}
void hungary(){for(i=1;i<=n;++i,memset(vis,0,sizeof(vis)))if(dfs(i))ans++;}
int main(){
	for(scanf("%d%d",&n,&m);m--;)scanf("%d%d",&i,&j),mp[i][j]=1;
	for(k=1;k<=n;++k)for(i=1;i<=n;++i)for(j=1;j<=n;++j)mp[i][j]|=mp[i][k]&mp[k][j];
	hungary();printf("%d\n",n-ans);
}
```

有向无环图最小路径点覆盖等于总点数 n-最大匹配数

```
#include<bits/stdc++.h>
using namespace std;const int N=57*57*4;
struct data{int to,next;}e[N];int cnt,head[N],vis[N],lnk[N],n,m,R,C,tot,i,j,ans,id[57][57];char s[57];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
bool dfs(int x){
	for(int i=head[x];i;i=e[i].next)if(!vis[e[i].to]){
		vis[e[i].to]=1;
		if(!lnk[e[i].to]||dfs(lnk[e[i].to])){lnk[e[i].to]=x;return true;}
	}return false;
}
void hungary(){for(ans=0,i=1;i<=tot;++i,memset(vis,0,sizeof(vis)))if(dfs(i))ans++;}
int main(){

for(scanf("%d%d%d%d",&n,&m,&R,&C),i=1;i<=n;++i)for(scanf("%s",s+1),j=1;j<=m;++j)if(s[j]=='.')id
[i][j]=++tot;
	for(i=1;i<=n;++i)for(j=1;j<=m;++j)if(id[i][j]){
		if(i+R<=n&&j+C<=m&&id[i+R][j+C])ins(id[i][j],id[i+R][j+C]);
		if(i+C<=n&&j+R<=m&&id[i+C][j+R])ins(id[i][j],id[i+C][j+R]);
		if(i+R<=n&&j-C>=1&&id[i+R][j-C])ins(id[i][j],id[i+R][j-C]);
		if(i+C<=n&&j-R>=1&&id[i+C][j-R])ins(id[i][j],id[i+C][j-R]);
	}
	hungary();printf("%d\n",tot-ans);
}
```

tarjan 求割点

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7,M=1e6+7;typedef long long ll;
```

```
struct data{int to,next;}e[M];int cnt,head[N],ind,dfn[N],low[N],size[N],cut[N],n,m,i,j;ll ans[N];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs(int x){
	dfn[x]=low[x]=++ind;int sum=0,flag=0,y;size[x]=1;
	for(int i=head[x];i;i=e[i].next){
		if(!dfn[y=e[i].to]){
			dfs(y);
			size[x]+=size[y];
			low[x]=min(low[x],low[y]);
			if(low[y]>=dfn[x]){
				flag++;
				sum+=size[y];
				if(x!=1||flag>1)cut[x]=1;
				ans[x]+=(ll)size[y]*(n-size[y]);
			}
		}else low[x]=min(dfn[y],low[x]);
	}
	if(cut[x])ans[x]+=(n-1)+ll(n-sum-1)*(sum+1);else ans[x]=2*(n-1);
}
int main(){
	for(scanf("%d%d",&n,&m);m--;){scanf("%d%d",&i,&j);if(i==j)continue;insert(i,j);}dfs(1);
	for(i=1;i<=n;++i)printf("%lld\n",ans[i]);
}
```

tarjan 求桥

```
#include<bits/stdc++.h>
using namespace std;const int N=4e4+7,inf=1e9+7;
struct data{int to,next;}e[N];int size[N],cnt=1,head[N],tot,n,m,i,j,dfn[N],low[N],ind,ans=inf;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs(int x,int fa=0){
	dfn[x]=low[x]=++ind;
	for(int i=head[x];i;i=e[i].next)if(!dfn[e[i].to]){
		dfs(e[i].to,i);
		size[x]+=size[e[i].to];
		low[x]=min(low[x],low[e[i].to]);
		if(dfn[x]<low[e[i].to])ans=min(ans,abs(tot-2*size[e[i].to]));
	}else if((i|1)!=(fa|1))low[x]=min(low[x],dfn[e[i].to]);
}
int main(){
```

```
for(;~scanf("%d%d",&n,&m);cnt=1,tot=0,ans=inf,memset(dfn,0,sizeof(dfn)),memset(head,0,sizeo
f(head))){
        for(i=1;i<=n;++i)scanf("%d",size+i),tot+=size[i];
        for(;m--;)scanf("%d%d",&i,&j),insert(i+1,j+1);
        dfs(1);if(ans==inf)puts("impossible");else printf("%d\n",ans);
    }
}
```

一条边是桥，当且仅当搜索树上 x 有一子节点 y dfn[x]<low[y]

一个点是割点，如果 x 不是搜索起始点，有个两个子节点点 y,满足 dfn[x]<=low[y]

求点双联通分量里边的数量

```
#include<bits/stdc++.h>
#define fr first
#define sc second
using namespace std;const int N=1e4+7,M=2e5+7;typedef pair<int,int>pa;
struct data{int to,next;}e[M];int
cnt=1,head[N],ind,top,bcc,bl[M*2],dfn[N],low[N],vis[N],tot,V[M],E[M],ans1,ans2,n,m,i,j;pa q[M];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs(int x,int fa=0){
    dfn[x]=low[x]=++ind;
    for(int i=head[x],y;i;i=e[i].next)if((y=e[i].to)!=fa){
        if(!dfn[y]){
            q[++top]=pa(x,y);
            dfs(y,x);
            low[x]=min(low[x],low[y]);
            if(dfn[x]<=low[y]){
                pa z(0,0);tot=0;bcc++;
                while(z!=pa(x,y))z=q[top--],bl[++tot]=z.fr,bl[++tot]=z.sc,E[bcc]++;
                for(int i=1;i<=tot;++i)if(!vis[bl[i]])vis[bl[i]]=1,V[bcc]++;
                for(int i=1;i<=tot;++i)vis[bl[i]]=0;
            }
        }else if(dfn[y]<dfn[x]){
            low[x]=min(low[x],dfn[y]);
            q[++top]=pa(x,y);
        }
    }
}
int main(){

for(;scanf("%d%d",&n,&m),n+m;top=cnt=bcc=ans1=ans2=ind=0,memset(dfn,0,sizeof(dfn)),mems
```

```
et(V,0,sizeof(V)),memset(E,0,sizeof(E)),memset(head,0,sizeof(head))){
        for(;m--;)scanf("%d%d",&i,&j),insert(i+1,j+1);
        for(i=1;i<=n;++i)if(!dfn[i])dfs(i);
        for(i=1;i<=bcc;++i)if(V[i]>E[i])ans1++;else if(E[i]>V[i])ans2+=E[i];
        printf("%d %d\n",ans1,ans2);
    }
}
```

求边双连通分量 先找到桥，然后并查集合并

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=4e5+7;typedef pair<int,int>pa;
struct data{int to,next,v;}e[N*2];int
cnt=1,fa[N],n,m,cut[N],low[N],from[N],dfn[N],d[N],ans,ind,i,j,kase;
int find(int x){return x==fa[x]?x:fa[x]=find(fa[x]);}
struct graph{
    int head[N];
    void init(){memset(head,0,sizeof(head));}
    void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
    void insert(int u,int v,int w=0){ins(u,v,w);ins(v,u,w);}
}g1,g2;
void dfs(int x,int id){
    dfn[x]=low[x]=++ind;
    for(int i=g1.head[x];i;i=e[i].next)if(i/2!=id/2)if(!dfn[e[i].to]){
            dfs(e[i].to,i);
            low[x]=min(low[e[i].to],low[x]);
            if(dfn[x]<low[e[i].to])cut[i/2]=1,ans++;
    }else low[x]=min(low[x],dfn[e[i].to]);
}
void dfs(int x){
    for(int i=g2.head[x];i;i=e[i].next)
        if(d[e[i].to]==-1)d[e[i].to]=d[x]+1,from[e[i].to]=i,dfs(e[i].to);
}
int main(){

for(;scanf("%d%d",&n,&m),n+m;g1.init(),g2.init(),ind=ans=0,cnt=1,memset(cut,0,sizeof(cut)),me
mset(dfn,0,sizeof(dfn))){
        for(i=1;i<=n;++i)fa[i]=i;printf("Case %d:\n",++kase);
        for(;m--;)scanf("%d%d",&i,&j),g1.insert(i,j);
```

```
        for(dfs(1,0),i=1;i<=n;++i)for(j=g1.head[i];j;j=e[j].next)if(!cut[j/2])fa[find(i)]=find(e[j].to);
        for(i=1;i<=n;++i)for(j=g1.head[i];j;j=e[j].next)if(cut[j/2])g2.insert(find(i),find(e[j].to),1);
        memset(d,-1,sizeof(d));memset(from,0,sizeof(from));d[find(1)]=0;dfs(find(1));
        for(scanf("%d",&m);m--;){
            scanf("%d%d",&i,&j);
            if(find(i)==find(j))printf("%d\n",ans);
            else{
                i=find(i);j=find(j);
                while(i!=j){
                    if(d[i]<d[j])swap(i,j);
                    if(e[from[i]].v)ans--,e[from[i]].v=0;
                    i=e[from[i]^1].to;
                }
                printf("%d\n",ans);
            }
        }
    }
}
```

无向图点双连通分量判断奇环，如果可以用二分图染色，则不存在基环，

```
#include<cstdio>
#include<cstring>
#include<algorithm>
#define fr first
#define sc second
using namespace std;const int N=1e3+7,M=N*N*2;typedef pair<int,int>pa;
struct data{int to,next;}e[M*4];int
mp[N][N],n,m,i,j,ans,flag,cnt,col[N],bl[M*2],vis[N],dfn[N],low[N],ind,top;pa q[M];
struct graph{
    int head[N];
    void init(){memset(head,0,sizeof(head));}
    void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
    void insert(int u,int v){ins(u,v);ins(v,u);}
}g1,g2;
void dfs(int x){
    if(flag)return;
    for(int i=g2.head[x];i;i=e[i].next)if(!col[e[i].to]){
        col[e[i].to]=-col[x];dfs(e[i].to);
    }else if(col[x]==col[e[i].to]){flag=1;return;}
}
void dfs(int x,int fa){
    dfn[x]=low[x]=++ind;
```

```
    for(int i=g1.head[x],y,tot=0;i;i=e[i].next)if((y=e[i].to)!=fa)if(!dfn[y]){
        q[++top]=pa(x,y);
        dfs(y,x);
        low[x]=min(low[x],low[y]);
        if(dfn[x]<=low[y]){
            tot=0;pa
z(0,0);while(z!=pa(x,y))z=q[top--],bl[++tot]=z.fr,bl[++tot]=z.sc,g2.insert(z.fr,z.sc);
            flag=0;col[x]=1;dfs(x);for(int i=1;i<=tot;++i)vis[bl[i]]|=flag;
            for(int i=1;i<=tot;++i)g2.head[bl[i]]=col[bl[i]]=0;
        }
    }else if(dfn[y]<dfn[x]){
        low[x]=min(low[x],dfn[y]);
        q[++top]=pa(x,y);
    }
}
int main(){

for(;scanf("%d%d",&n,&m),n+m;memset(vis,0,sizeof(vis)),memset(mp,0,sizeof(mp)),g1.init(),me
mset(dfn,0,sizeof(dfn)),ind=top=ans=cnt=0){
        for(;m--;)scanf("%d%d",&i,&j),mp[i][j]=mp[j][i]=1;
        for(i=1;i<=n;++i)for(j=i+1;j<=n;++j)if(!mp[i][j])g1.insert(i,j);
        for(i=1;i<=n;++i)if(!dfn[i])dfs(i,0);
        for(i=1;i<=n;++i)if(!vis[i])ans++;printf("%d\n",ans);
    }
}
```

把无向图的边拆成两条方向相反的有向边，做欧拉回路。

欧拉回路做法：

1、起点入栈；（回路的话起点可以是任意的）

2、扫描与起点相连的所有未被标记的边，对每条这样的边都标记它，然后它的终点入栈，递归处理；

3、如果从某个结点出发没有未被标记的边，则把这个结点出栈，加入答案序列中；

4、重复以上步骤，直到栈空；

5、对无向图，倒序的答案序列是一条欧拉回路，有向图正序倒序均可。

```
#include<cstdio>
const int N=1e5+7;
struct data{int to,next;}e[N];int cnt,head[N],vis[N],n,m,i,j;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
```

```
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs(int x){
	for(int i=head[x];i;i=e[i].next)if(!vis[i])vis[i]=1,dfs(e[i].to);printf("%d\n",x);
}
int main(){
	for(scanf("%d%d",&n,&m);m--;)scanf("%d%d",&i,&j),insert(i,j);dfs(1);
}
```

tarjan 算法有向图强联通分量

```
#include<cstdio>
#include<algorithm>
using namespace std;const int N=1e2+7,M=N*N;
struct data{int to,next;}e[M];int
cnt,head[N],q[N],top,inq[N],bl[N],dfn[N],low[N],scc,out[N],in[N],ans1,ans2,ind,n,i,j;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void dfs(int x){
	dfn[x]=low[x]=++ind;inq[x]=1;q[++top]=x;
	for(int i=head[x];i;i=e[i].next)if(!dfn[e[i].to])dfs(e[i].to),low[x]=min(low[x],low[e[i].to]);
	else if(inq[e[i].to])low[x]=min(low[x],dfn[e[i].to]);
	if(low[x]==dfn[x]){
		int now=-1;scc++;
		while(now!=x){
			now=q[top--];
			bl[now]=scc;
			inq[now]=0;
		}
	}
}
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)for(;scanf("%d",&j),j;)ins(i,j);
	for(i=1;i<=n;++i)if(!dfn[i])dfs(i);
	for(i=1;i<=n;++i)for(j=head[i];j;j=e[j].next)if(bl[e[j].to]!=bl[i])out[bl[i]]++,in[bl[e[j].to]]++;
	for(i=1;i<=scc;++i){
		if(in[i]==0)ans1++;
		if(out[i]==0)ans2++;
	}
	printf("%d\n%d\n",ans1,scc==1?0:max(ans1,ans2));
}
```

bitset 优化的 kosaraju 算法

```
#include<cstdio>
#include<bitset>
using namespace std;const int N=107;typedef bitset<N> bit;
bit vis,s1[N],s2[N],s3[N];int n,i,j,k,q[N],b[N],top,in[N],out[N],bl[N],scc,ans1,ans2;
void dfs(int x,bit s[N]){
	vis[x]=1;
	for(;s[x]&=~vis,s[x].any();)dfs(s[x]._Find_first(),s);
	q[++top]=x;
}
void kosaraju(){
	top=0;for(i=1;i<=n;++i)if(!vis[i])dfs(i,s1);
	for(i=1;i<=n;++i)b[i]=q[i];vis.reset();
	for(i=n;i>=1;--i)if(!vis[b[i]]){
		top=0;scc++;dfs(b[i],s2);for(j=1;j<=top;++j)bl[q[j]]=scc;
	}
	for(i=1;i<=n;++i)for(j=1;j<=n;++j)if(s3[i][j]&&bl[i]!=bl[j])out[bl[i]]++,in[bl[j]]++;
	for(i=1;i<=scc;++i){
		if(!in[i])ans1++;
		if(!out[i])ans2++;
	}printf("%d\n%d\n",ans1,scc==1?0:max(ans1,ans2));
}
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)for(;scanf("%d",&j),j;)s1[i][j]=s3[i][j]=s2[j][i]=1;kosaraju();
}
```

2 sat 建图　如果必须选 A,那么 A==true,建边<i',i>　如果 belong[i]<belong[i']　那么 i 为真，否则 i 为假

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=2*1e3+7,M=1e7;
struct data{int to,next;}e[M];int
cnt,n,m,i,head[N],dfn[N],bl[N],inq[N],low[N],ind,scc,a,b,c,q[N],top;char s[20];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void dfs(int x){
	dfn[x]=low[x]=++ind;q[++top]=x;inq[x]=1;
	for(int i=head[x];i;i=e[i].next)if(!dfn[e[i].to]){
		dfs(e[i].to);low[x]=min(low[e[i].to],low[x]);
	}else if(inq[e[i].to])low[x]=min(low[x],dfn[e[i].to]);
	if(low[x]==dfn[x]){
		int now=-1;scc++;
```

```
            while(now!=x){
                now=q[top--];
                bl[now]=scc;
                inq[now]=0;
            }
        }
}
int main(){
    for(scanf("%d%d",&n,&m),i=1;i<=m;++i){
        scanf("%d%d%d%s",&a,&b,&c,s);a++;b++;
        if(!strcmp(s,"XOR")){
            if(c==0){
                ins(a,b);
                ins(b,a);
                ins(a+n,b+n);
                ins(b+n,a+n);
            }else{
                ins(a,b+n);
                ins(b+n,a);
                ins(b,a+n);
                ins(a+n,b);
            }
        }
        if(!strcmp(s,"OR")){
            if(c==0){
                ins(a,a+n);
                ins(b,b+n);
            }else{
                ins(a+n,b);
                ins(b+n,a);
            }
        }
        if(!strcmp(s,"AND")){
            if(c==0){
                ins(a,b+n);
                ins(b,a+n);
            }else{
                ins(a+n,a);
                ins(b+n,b);
            }
        }
    }
    for(i=1;i<=n*2;++i)if(!dfn[i])dfs(i);
    for(i=1;i<=n;++i)if(bl[i]==bl[i+n])return 0*puts("NO");
```

```
	puts("YES");
}
```

******仙人掌图求直径

```
#include<bits/stdc++.h>
using namespace std;const int N=5e4+7,M=1e6+7;
struct data{int to,next;}e[M<<1];int n,m,i,j,head[N],f[N],ans,tot,x,y,cnt,q[N*2],t,w,a[N*2],ind,d[N],fa[N],dfn[N],low[N];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dp(int root,int x){
	for(tot=d[x]-d[root]+1,i=x;i!=root;i=fa[i])a[tot--]=f[i];
	a[1]=f[root];
	for(tot=d[x]-d[root]+1,i=1;i<=tot;++i)a[i+tot]=a[i];
	for(q[0]=1,t=0,w=1,i=2;i<=tot*2;++i){
		while(t<w&&i-q[t]>tot/2)t++;
		ans=max(ans,a[i]+i+a[q[t]]-q[t]);
		while(t<w&&a[q[w-1]]-q[w-1]<=a[i]-i)w--;
		q[w++]=i;
	}
	for(i=2;i<=tot;++i)
		f[root]=max(f[root],a[i]+min(i-1,tot-i+1));
}
void dfs(int x){
	dfn[x]=low[x]=++ind;
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
		if(!dfn[e[i].to]){
			fa[e[i].to]=x;
			d[e[i].to]=d[x]+1;
			dfs(e[i].to);
			low[x]=min(low[x],low[e[i].to]);
		}else low[x]=min(low[x],dfn[e[i].to]);
		if(dfn[x]<low[e[i].to]){
			ans=max(ans,f[x]+f[e[i].to]+1);
			f[x]=max(f[x],f[e[i].to]+1);
		}
	}
	for(int i=head[x];i;i=e[i].next)
		if(fa[e[i].to]!=x&&dfn[x]<dfn[e[i].to])
			dp(x,e[i].to);
}
int main(){
	for(scanf("%d%d",&n,&m);m--;)for(scanf("%d%d",&j,&x),j--;j--;x=y)scanf("%d",&y),insert(x,y);
```

```
	dfs(1);printf("%d\n",ans);
}
```

堆树

```
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<algorithm>
using namespace std;const int N=5e4+7;
int c[N][2],v[N],rnd[N],sz,i,j,q[N],top,st,ed,n,m,root;char ch;
void rotate(int&y,int l){
	int x=c[y][l],r=l^1;
	c[y][l]=c[x][r];c[x][r]=y;y=x;
}
void insert(int&x,int val){
	if(!x){x=++sz;v[x]=val;rnd[x]=rand();return;}
	if(val<v[x]){
		insert(c[x][0],val);
		if(rnd[c[x][0]]<rnd[x])rotate(x,0);
	}else{
		insert(c[x][1],val);
		if(rnd[c[x][1]]<rnd[x])rotate(x,1);
	}
}
void del(int&x,int val){
	if(v[x]==val){
		if(!c[x][0]||!c[x][1])x=c[x][0]+c[x][1];
		else if(rnd[c[x][0]]<rnd[c[x][1]]){
			rotate(x,0);del(x,val);
		}else{
			rotate(x,1);del(x,val);
		}
	}else if(val<v[x])del(c[x][0],val);else del(c[x][1],val);
}
void ask(int x,int val){
	if(!x)return;
	if(v[x]<=val)st=max(st,v[x]);
	if(v[x]>=val)ed=min(ed,v[x]);
	if(val<=v[x])ask(c[x][0],val);else ask(c[x][1],val);
}
int main(){
```

```
	for(scanf("%d%d",&n,&m);m--;){
		scanf(" %c",&ch);
		if(ch=='D')scanf("%d",&q[++top]),insert(root,q[top]);
		if(ch=='R')del(root,q[top--]);
		if(ch=='Q'){
			scanf("%d",&i);st=0;ed=n+1;ask(root,i);
			if(st==i||ed==i)puts("0");else printf("%d\n",ed-st-1);
		}
	}
}
```

非旋转 treap

```
#include<cstdio>
#include<cstdlib>
#include<algorithm>
using namespace std;const int N=5e4+7;
int ls[N],rs[N],rnd[N],v[N],sz,i,j,root,x,y,a,b,st,ed,top,n,m,q[N];char ch;
int merge(int x,int y){
	if(!x||!y)return x+y;int z=0;
	if(rnd[x]<rnd[y])z=x,rs[z]=merge(rs[x],y);
	else z=y,ls[z]=merge(x,ls[y]);
	return z;
}
void split(int x,int&y,int &z,int val){
	y=z=0;if(!x)return;
	if(v[x]<=val)y=x,split(rs[x],rs[y],z,val);
	else z=x,split(ls[x],y,ls[z],val);
}
void ask(int x,int val){
	if(!x)return;
	if(v[x]<=val)st=max(st,v[x]);
	if(v[x]>=val)ed=min(ed,v[x]);
	if(val<v[x])ask(ls[x],val);else ask(rs[x],val);
}
int main(){
	for(scanf("%d%d",&n,&m);m--;){
		scanf(" %c",&ch);
		if(ch=='D'){
			scanf("%d",&q[++top]);split(root,x,y,q[top]);v[++sz]=q[top];rnd[sz]=rand();
			root=merge(merge(x,sz),y);
		}
```

```
            if(ch=='R'){
                split(root,x,y,q[top]);split(x,a,b,q[top]-1);top--;
                root=merge(a,y);
            }
            if(ch=='Q'){
                scanf("%d",&i);st=0;ed=n+1;ask(root,i);
                if(st==i||ed==i)puts("0");else printf("%d\n",ed-st-1);
            }
        }
}
```

伸展树

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;
int c[N][2],v[N],fa[N],sz,i,l,r,x,y,z,n,m,root,q[N],top,rev[N],size[N];
void rever(int x){if(!x)return;rev[x]^=1;swap(c[x][0],c[x][1]);}
void pushdown(int x){if(rev[x])rever(c[x][0]),rever(c[x][1]),rev[x]=0;}
void update(int x){size[x]=size[c[x][0]]+size[c[x][1]]+1;}
void rotate(int x,int&k){
    int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
    if(y==k)k=x;else c[z][c[z][1]==y]=x;
    fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
    c[y][l]=c[x][r];c[x][r]=y;
    update(y);update(x);
}
void splay(int x,int&k){
    while(x!=k){
        int y=fa[x],z=fa[y];
        if(y!=k){
            if(c[y][0]==x^c[z][0]==y)rotate(x,k);else rotate(y,k);
        }rotate(x,k);
    }
}
void build(int l,int r,int f){
    if(l>r)return;int x=l+r>>1;fa[x]=f;size[x]=1;
    if(x<f)c[f][0]=x;else c[f][1]=x;if(l==r)return;
    build(l,x-1,x);build(x+1,r,x);update(x);
}
int find(int x,int rnk){
    pushdown(x);
    if(size[c[x][0]]+1==rnk)return x;
    else if(size[c[x][0]]>=rnk)return find(c[x][0],rnk);
```

```
    else return find(c[x][1],rnk-size[c[x][0]]-1);
}
int main(){
    for(scanf("%d%d",&n,&m),root=(3+n)>>1,build(1,n+2,0);m--;){
        scanf("%d%d",&l,&r);
        x=find(root,l);y=find(root,r+2);splay(x,root);splay(y,c[x][1]);z=c[y][0];rever(z);
    }
    for(i=1;i<=n;++i)printf("%d%c",find(root,i+1)-1,i==n?'\n':' ');
}
```

非旋转 treap 解决区间问题

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;
int rnd[N],ls[N],rs[N],rev[N],size[N],last,sz,n,m,i,l,r,x,y,a,b,top,root,q[N];
void rever(int x){if(!x)return;rev[x]^=1;swap(rs[x],ls[x]);}
void update(int x){size[x]=size[ls[x]]+size[rs[x]]+1;}
void pushdown(int x){if(rev[x])rever(ls[x]),rever(rs[x]),rev[x]=0;}
int merge(int x,int y){
    if(!x||!y)return x+y;int z=0;
    if(rnd[x]<rnd[y])z=x,pushdown(z),rs[z]=merge(rs[x],y);
    else z=y,pushdown(z),ls[z]=merge(x,ls[y]);update(z);return z;
}
void split(int x,int&y,int&z,int k){
    y=z=0;if(!x)return;pushdown(x);
    if(size[ls[x]]>=k)z=x,split(ls[x],y,ls[z],k),update(z);
    else y=x,split(rs[x],rs[y],z,k-size[ls[x]]-1),update(y);
}
int build(){
    for(i=1;i<=n;++i){
        last=0;size[i]=1;rnd[i]=rand();
        while(top&&rnd[q[top]]>rnd[i])update(last=q[top--]);
        if(top)rs[q[top]]=i;
        ls[i]=last;
        q[++top]=i;
    }while(top)update(q[top--]);return q[1];
}
void dfs(int x){if(!x)return;pushdown(x);dfs(ls[x]);printf("%d ",x);dfs(rs[x]);}
int main(){
    for(scanf("%d%d",&n,&m),root=build();m--;){
        scanf("%d%d",&l,&r);split(root,x,y,l-1);split(y,a,b,r-l+1);rever(a);
        root=merge(x,merge(a,b));
    }dfs(root);
```

```
}
```

ETT

```
#include<bits/stdc++.h>
using namespace std;const int N=4e5+7;
int n,m,mod,d[N],i,j,k,c[N][2],v[N],sz,fa[N],size[N],w[N][2],L[N],R[N],h[N];
struct data{int to,next;}e[N];int head[N],cnt,a[N],root,op,x,y,z,id,val,tag,l,r;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void dfs(int x){
	L[x]=++sz;v[sz]=a[x];d[sz]=h[x];
	for(int i=head[x];i;i=e[i].next)h[e[i].to]=h[x]+1,dfs(e[i].to);
	R[x]=++sz;
}
void update(int x){
	size[x]=size[c[x][0]]+size[c[x][1]]+1;
	w[x][0]=w[c[x][0]][0]^w[c[x][1]][0];
	w[x][1]=w[c[x][0]][1]^w[c[x][1]][1];
	w[x][d[x]&1]^=v[x];
}
void rotate(int x,int&k){
	int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
	if(y==k)k=x;else c[z][c[z][1]==y]=x;
	fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
	c[y][l]=c[x][r];c[x][r]=y;
	update(y);update(x);
}
void splay(int x,int&k){
	while(x!=k){
		int y=fa[x],z=fa[y];
		if(y!=k){
			if(c[y][0]==x^c[z][0]==y)rotate(x,k);else rotate(y,k);
		}rotate(x,k);
	}
}
int find(int x,int rnk){
	if(size[c[x][0]]+1==rnk)return x;
	else if(size[c[x][0]]>=rnk)return find(c[x][0],rnk);
	else return find(c[x][1],rnk-size[c[x][0]]-1);
}
void build(int l,int r,int f){
	if(l>r)return;int x=l+r>>1;fa[x]=f;
	if(x<f)c[f][0]=x;else c[f][1]=x;if(l==r){update(x);return;}
```

```
	build(l,x-1,x);build(x+1,r,x);update(x);
}
int main(){
	for(scanf("%d%d",&n,&mod),++mod,i=1;i<=n;++i)scanf("%d",a+i),a[i]%=mod;
	for(i=1;i<n;++i)scanf("%d%d",&j,&k),ins(j,k);
	dfs(1);build(1,sz,0);root=(1+sz)>>1;
	for(scanf("%d",&m);m--;){
		scanf("%d",&op);
		if(op==1){
			scanf("%d",&i);i^=tag;z=h[i]&1;x=L[i];y=R[i];
			splay(x,root);splay(y,c[x][1]);
			if(w[c[y][0]][z^1])puts("MeiZ"),tag++;else puts("GTY");
		}

if(op==2){scanf("%d%d",&l,&r);l^=tag;r^=tag;splay(L[l],root);v[root]=r%mod;update(root);}
		if(op==3){
			scanf("%d%d%d",&x,&id,&val);x^=tag;id^=tag;val^=tag;
			L[id]=++sz;v[sz]=val%mod;h[id]=d[sz]=h[x]+1;
			splay(L[x],root);x=root;
			y=find(c[x][1],1);splay(y,c[x][1]);
			c[y][0]=sz;fa[sz]=y;
			R[id]=++sz;fa[sz]=sz-1;c[sz-1][1]=sz;
			update(sz);update(sz-1);update(y);update(x);
		}
	}
}
```

矩阵树定理

```
#include<bits/stdc++.h>
using namespace std;const int N=20;const double eps=1e-8;
double a[N][N],t,ans;int n,m,i,j,k,l,T_T;
void gauss(){
	for(i=l=1;i<=n&&l<=n;++l){
		for(j=i;j<=n;++j)if(fabs(a[j][l])>eps)break;if(j==n+1)continue;
		for(k=1;k<=n;++k)swap(a[i][k],a[j][k]);
		for(j=i+1;j<=n;++j)
			for(t=a[j][l]/a[i][l],k=1;k<=n;++k)a[j][k]-=t*a[i][k];++i;
	}
	for(ans=n&1?-1:1,i=1;i<=n;++i)ans*=a[i][i];printf("%.0f\n",fabs(ans));
}
int main(){
```

```
    for(scanf("%d",&T_T);T_T--;gauss(),memset(a,0,sizeof(a))){
        for(scanf("%d%d",&n,&m),n--;m--;)
            scanf("%d%d",&i,&j),i--,j--,a[i][i]++,a[j][j]++,a[i][j]--,a[j][i]--;
    }
}
```

有向图矩阵树定理

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;const ll N=307,mod=1e9+7;
char s[N];ll a[N][N],ans;int n,i,j,k,l,t,inv,f;
void add(ll&x,ll v){v%=mod;x+=v;x=(x%mod+mod)%mod;}
ll pow_mod(ll x,ll n){
    ll res=1;
    while(n){
        if(n&1)res=res*x%mod;
        x=x*x%mod;
        n>>=1;
    }return res;
}
void gauss(){
    for(i=l=1,f=1;i<=n&&l<=n;++l){
        for(j=i;j<=n;++j)if(a[j][l])break;if(j==n+1)continue;
        if(j!=i){
            for(k=1;k<=n;++k)swap(a[i][k],a[j][k]);f=-f;
        }
        for(j=i+1;j<=n;++j)
            for(t=a[j][l]*pow_mod(a[i][l],mod-2)%mod,k=1;k<=n;++k)add(a[j][k],-t*a[i][k]);++i;
    }
    for(ans=1,i=1;i<=n;++i)ans=ans*a[i][i]%mod;printf("%lld\n",(mod+f)%mod*ans);
}
int main(){
    for(scanf("%d",&n),i=1;i<=n;++i)for(scanf("%s",s+1),j=1;j<=n;++j)
        if(s[j]=='1')add(a[j-1][j-1],1),add(a[i-1][j-1],mod-1);
    n--;gauss();
}
```

矩阵树加辗转相除

```
#include<bits/stdc++.h>
using namespace std;const int N=107,mod=1e9;typedef long long ll;
```

```
int id[N][N],n,m,i,j,k,l,f,cnt,dx[4]={0,0,1,-1},dy[4]={1,-1,0,0},x,y;char s[N][N];ll a[N][N],t,ans;
void ins(int u,int v){a[v][v]++;a[u][v]=(a[u][v]+mod-1)%mod;}
void gauss(){
	for(f=1,i=l=1;i<=n&&l<=n;++l){
		for(j=i;j<=n;++j)if(a[j][l])break;if(j==n+1)continue;
		if(j!=i){for(k=1;k<=n;++k)swap(a[i][k],a[j][k]);f=-f;}
		for(j=i+1;j<=n;++j)while(a[j][l]){
			for(k=1,t=a[j][l]/a[i][l];k<=n;++k)a[j][k]=((a[j][k]-t*a[i][k])%mod+mod)%mod;
			if(a[j][l]){
				for(k=1;k<=n;++k)swap(a[i][k],a[j][k]);f=-f;
			}
		}++i;
	}
	for(ans=i=1;i<=n;++i)ans=ans*a[i][i]%mod;printf("%lld\n",f==1?ans:(mod-ans));
}
int main(){

for(scanf("%d%d",&n,&m),i=1;i<=n;++i)for(scanf("%s",s[i]+1),j=1;j<=m;++j)if(s[i][j]=='.')id[i][j]=++c
nt;
	for(i=1;i<=n;++i)for(j=1;j<=m;++j)if(id[i][j])for(k=0;k<4;++k){
		x=dx[k]+i;y=dy[k]+j;
		if(x>=1&&x<=n&&y>=1&&y<=m&&id[x][y])ins(id[i][j]-1,id[x][y]-1);
	}
	n=cnt-1;gauss();
}
```

概率矩阵树

```
#include<bits/stdc++.h>
using namespace std;const double eps=1e-9;const int N=57;
double a[N][N],f,tot,t,ans;int n,m,i,j,k,l;
void ins(int x,int y,double z){a[x][y]-=z;a[y][y]+=z;}
void gauss(){
	for(i=l=1;l<=n&&i<=n;++l){
		for(k=j=i;k<=n;++k)if(fabs(a[k][l])>fabs(a[j][l]))j=k;if(fabs(a[j][l])<eps)continue;
		for(k=1;k<=n;++k)swap(a[i][k],a[j][k]);
		for(j=i+1;j<=n;++j)for(t=a[j][l]/a[i][l],k=1;k<=n;++k)a[j][k]-=t*a[i][k];++i;
	}
	for(ans=i=1;i<=n;++i)ans*=a[i][i];printf("%.9f\n",fabs(ans*tot));
}
int main(){
	for(scanf("%d",&n),tot=i=1;i<=n;++i)for(j=1;j<=n;++j){
```

```
        scanf("%lf",&f);if(fabs(f-1)<=eps)f=1-eps;if(i>=j)continue;tot*=1-f;
        ins(i-1,j-1,f/(1-f));
        ins(j-1,i-1,f/(1-f));
    }
    n--;gauss();
}
```

矩阵树加容斥

```
#include<bits/stdc++.h>
using namespace std;const int N=20,mod=1e9+7;typedef pair<int,int>pa;typedef long long ll;
ll pow_mod(ll x,ll n){
    ll res=1;
    while(n){
        if(n&1)res=res*x%mod;
        x=x*x%mod;
        n>>=1;
    }return res;
}
int n,m,i,j,k,l,x,y,kase,cnt,f=1;vector<pa>s[N];ll a[N][N],b[N][N],ans,res,t,inv;
void ins(int x,int y){a[x][y]=(a[x][y]+(mod-1))%mod;a[y][y]++;}
void insert(int x,int y){ins(x,y);ins(y,x);}
ll gauss(){
    for(i=l=f=1;i<=n&&l<=n;++l){
        for(j=i;j<=n;++j)if(a[j][l])break;if(j==n+1)continue;
        if(j!=i)for(f=-f,k=1;k<=n;++k)swap(a[i][k],a[j][k]);inv=pow_mod(a[i][l],mod-2);
        for(j=i+1;j<=n;++j)
            for(k=1,t=a[j][l]*inv%mod;k<=n;++k)a[j][k]=(a[j][k]-a[i][k]*t%mod+mod)%mod;++i;
    }
    for(res=i=1;i<=n;++i)res=(res*a[i][i])%mod;return f==1?res:(mod-res);
}
int main(){

for(scanf("%d",&n),n--,i=0;i<n;++i)for(scanf("%d",&j);j--;)scanf("%d%d",&x,&y),s[i].push_back(pa(
x,y));
    for(kase=0;kase<(1<<n);memcpy(a,b,sizeof(b)),++kase){
        for(i=cnt=0;i<n;++i)if(kase>>i&1)for(cnt++,j=0;j<s[i].size();++j)
            insert(s[i][j].first-1,s[i][j].second-1);cnt=n-cnt;
        ans=((ans+(cnt&1?-1:1)*gauss())%mod+mod)%mod;
    }printf("%lld\n",ans);
}
```

带修改莫队 block=(n)^(2/3)

```
#include<bits/stdc++.h>
using namespace std;const int N=5e4+7;
struct data{int l,r,tim,id;}e[N],c[N];int
res[N],block,n,i,j,T_T,L,R,vis[N],a[N],tot,cnt,ind,ans,kase;char s[10];
bool operator<(data a,data b){
    if(a.l/block==b.l/block){
        if(a.r/block==b.r/block)return a.tim<b.tim;
        else return a.r<b.r;
    }else return a.l<b.l;
}
void add(int x,int v){ans+=x*v;}
void change(int x,int val){if(vis[x])add(a[x],-1);a[x]+=val;if(vis[x])add(a[x],1);}
void go(int x){vis[x]^=1;add(a[x],vis[x]?1:-1);}
int main(){

for(scanf("%d",&T_T),kase=1;kase<=T_T;++kase,ans=ind=cnt=tot=0,memset(vis,0,sizeof(vis))){
        for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",a+i);block=pow(n+8,2.0/3);
        for(;scanf("%s",s),strcmp(s,"End");++ind){

if(!strcmp(s,"Query"))++cnt,scanf("%d%d",&e[cnt].l,&e[cnt].r),e[cnt].tim=ind,e[cnt].id=cnt;
            if(!strcmp(s,"Add"))++tot,scanf("%d%d",&i,&j),c[tot].l=i,c[tot].r=j,c[tot].tim=ind;
            if(!strcmp(s,"Sub"))++tot,scanf("%d%d",&i,&j),c[tot].l=i,c[tot].r=-j,c[tot].tim=ind;
        }
        for(sort(e+1,e+cnt+1),L=1,R=0,j=i=1;i<=cnt;++i){
            while(e[i].tim>c[j].tim&&j<=tot)change(c[j].l,c[j].r),j++;
            while(j>1&&e[i].tim<c[j-1].tim)j--,change(c[j].l,-c[j].r);
            while(R<e[i].r)go(++R);
            while(R>e[i].r)go(R--);
            while(L<e[i].l)go(L++);
            while(L>e[i].l)go(--L);
            res[e[i].id]=ans;
        }
        for(printf("Case %d:\n",kase),i=1;i<=cnt;++i)printf("%d\n",res[i]);
    }
}
```

树上莫队加带修改莫队

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;
struct Q{int l,r,tim,id;}s[N],ss[N];int tot0,tot1,op,x,y;typedef long long ll;ll ans,res[N],V[N],W[N];
```

```
struct data{int to,next;}e[N<<1];int cnt,fa[N][20],a[N],b[N],head[N],d[N],ind,i,j,k,q[N],top,bl[N],block,tot,vis[N],n,m,K;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
bool operator<(Q a,Q b){
	if(bl[a.l]==bl[b.l]){
		if(bl[a.r]==bl[b.r])return a.tim<b.tim;
		else return bl[a.r]<bl[b.r];
	}else return bl[a.l]<bl[b.l];
}
int dfs(int x){
	int size=0;
	for(int i=1;(1<<i)<=d[x];++i)fa[x][i]=fa[fa[x][i-1]][i-1];
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x][0]){
		fa[e[i].to][0]=x;
		d[e[i].to]=d[x]+1;
		size+=dfs(e[i].to);
		if(size>=block){
			tot++;
			for(int i=0;i<size;++i)bl[q[top--]]=tot;size=0;
		}
	}
	q[++top]=x;return size+1;
}
int lca(int u,int v){
	if(d[u]<d[v])swap(u,v);
	int t=d[u]-d[v];
	for(int i=19;i>=0;--i)if(t>>i&1)u=fa[u][i];
	if(u==v)return v;
	for(int i=19;i>=0;--i)if(fa[u][i]!=fa[v][i])u=fa[u][i],v=fa[v][i];
	return fa[u][0];
}
void reverse(int x){
	vis[x]^=1;
	if(vis[x]){
		b[a[x]]++;
		ans+=W[b[a[x]]]*V[a[x]];
	}else{
		ans-=W[b[a[x]]]*V[a[x]];
		b[a[x]]--;
	}
}
void change(int x,int v){
	if(vis[x]){
```

```
            ans-=W[b[a[x]]]*V[a[x]];
            b[a[x]]--;
            a[x]=v;
            b[a[x]]++;
            ans+=W[b[a[x]]]*V[a[x]];
        }else a[x]=v;
}
void go(int u,int v){while(u!=v){if(d[u]<d[v])swap(u,v);reverse(u);u=fa[u][0];}}
int main(){
        for(scanf("%d%d%d",&n,&m,&K),i=1;i<=m;++i)scanf("%lld",V+i);d[0]=-1;
        for(i=1;i<=n;++i)scanf("%lld",W+i);block=pow(n,2.0/3)*0.5;
        for(i=1;i<n;++i)scanf("%d%d",&j,&k),insert(j,k);
        for(dfs(1),++tot,i=1;i<=top;++i)bl[q[i]]=tot;
        for(i=1;i<=n;++i)scanf("%d",a+i);
        for(i=1;i<=K;++i){
            scanf("%d%d%d",&op,&x,&y);
            if(op==0)++tot0,ss[tot0].l=x,ss[tot0].r=a[x],ss[tot0].id=y,ss[tot0].tim=i,a[x]=y;
            if(op==1){
                if(bl[x]>bl[y])swap(x,y);
                ++tot1,s[tot1].l=x,s[tot1].r=y,s[tot1].id=tot1,s[tot1].tim=i;
            }
        }
        for(sort(s+1,s+tot1+1),j=tot0+1,i=1;i<=tot1;++i){
            while(j<=tot0&&ss[j].tim<s[i].tim)change(ss[j].l,ss[j].id),++j;
            while(j>1&&ss[j-1].tim>s[i].tim)j--,change(ss[j].l,ss[j].r);
            go(s[i].l,s[i-1].l);go(s[i].r,s[i-1].r);x=lca(s[i].l,s[i].r);
            reverse(x);res[s[i].id]=ans;reverse(x);
        }
        for(i=1;i<=tot1;++i)printf("%lld\n",res[i]);
}
```

回滚莫队

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;typedef long long ll;
struct data{int l,r,id;}e[N];int l,r,n,m,i,j,pos,block,L[N],R[N],bl[N],a[N],d[N];ll
sum[N],tmp[N],ans,last,res[N],cur;
bool operator<(data a,data b){return bl[a.l]==bl[b.l]?a.r<b.r:a.l<b.l;}
void add(int x,ll&ans,ll sum[N]){sum[x]+=d[x];ans=max(ans,sum[x]);}
void del(int x,ll sum[N]){sum[x]-=d[x];}
int main(){
        for(scanf("%d%d",&n,&m),block=sqrt(n+1),i=1;i<=n;++i){
            scanf("%d",a+i);d[i]=a[i];
```

```
        bl[i]=i/block+1;
        if(!L[bl[i]])L[bl[i]]=i;R[bl[i]]=i;
    }
    for(sort(d+1,d+n+1),i=1;i<=n;++i)a[i]=lower_bound(d+1,d+n+1,a[i])-d;
    for(i=1;i<=m;++i)scanf("%d%d",&e[i].l,&e[i].r),e[i].id=i;
    for(sort(e+1,e+m+1),i=1;i<=m;++i){
        if(bl[e[i].l]!=bl[e[i-1].l]){
            memset(sum,0,sizeof(sum));
            r=R[bl[e[i].l]];pos=l=r+1;ans=last=0;
        }
        if(bl[e[i].l]==bl[e[i].r]){
            cur=0;
            for(j=e[i].l;j<=e[i].r;++j)add(a[j],cur,tmp);res[e[i].id]=cur;
            for(j=e[i].l;j<=e[i].r;++j)del(a[j],tmp);continue;
        }
        while(r<e[i].r)add(a[++r],ans,sum);
        last=ans;
        while(l>e[i].l)add(a[--l],ans,sum);
        res[e[i].id]=ans;
        while(l<pos)del(a[l++],sum);
        ans=last;
    }for(i=1;i<=m;++i)printf("%lld\n",res[i]);
}
```

ETT 2

```
#include<bits/stdc++.h>
using namespace std;const int N=2e5+7;typedef long long ll;
int read(){
    int ret=0;char gc;
    while(gc<'0'||gc>'9')       gc=getchar();
    while(gc>='0'&&gc<='9')       ret=ret*10+gc-'0',gc=getchar();
    return ret;
}
int c[N][2],fa[N],size[N],a[N],w[N],len[N],n,m,i,j,k,sz,x,y,z,l,r;ll sum[N],v[N],tag[N];char ch;
struct data{int to,next;}e[N];int head[N],cnt,L[N],R[N],q[N],top,root;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void add(int x,ll val){if(!x)return;sum[x]+=len[x]*val;tag[x]+=val;v[x]+=w[x]*val;}
void pushdown(int x){if(tag[x])add(c[x][0],tag[x]),add(c[x][1],tag[x]),tag[x]=0;}
void update(int x){
    size[x]=size[c[x][0]]+size[c[x][1]]+1;
    len[x]=len[c[x][0]]+len[c[x][1]]+w[x];
    sum[x]=sum[c[x][0]]+sum[c[x][1]]+v[x];
```

```
}
void rotate(int x,int&k){
	int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
	if(y==k)k=x;else c[z][c[z][1]==y]=x;
	fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
	c[y][l]=c[x][r];c[x][r]=y;
	update(y);update(x);
}
void splay(int x,int&k){
	top=0;for(int i=x;i!=k;i=fa[i])q[++top]=i;q[++top]=k;
	while(top)pushdown(q[top--]);
	while(x!=k){
		int y=fa[x],z=fa[y];
		if(y!=k){
			if(c[y][0]==x^c[z][0]==y)rotate(x,k);else rotate(y,k);
		}rotate(x,k);
	}
}
void build(int l,int r,int f){
	if(l>r)return;int x=l+r>>1;fa[x]=f;
	if(x<f)c[f][0]=x;else c[f][1]=x;if(l==r){update(x);return;}
	build(l,x-1,x);build(x+1,r,x);update(x);
}
int find(int x,int rnk){
	if(size[c[x][0]]+1==rnk)return x;
	else if(size[c[x][0]]>=rnk)return find(c[x][0],rnk);
	else return find(c[x][1],rnk-size[c[x][0]]-1);
}
int qrank(int x,int ret=0){
	top=0;for(int i=x;i!=root;i=fa[i])q[++top]=i;q[++top]=root;
	for(int i=top;i>=1;--i){
		if(i==1)ret+=size[c[q[i]][0]]+1;
		else{
			if(c[q[i]][1]==q[i-1])ret+=size[q[i]]-size[q[i-1]];
		}
	}return ret;
}
void dfs(int x){
	L[x]=++sz;v[sz]=a[x];w[sz]=1;
	for(int i=head[x];i;i=e[i].next)dfs(e[i].to);
	R[x]=++sz;v[sz]=-a[x];w[sz]=-1;
}
void work(){
	l=qrank(L[i])-1;r=qrank(R[i])+1;
```

```
    x=find(root,l);y=find(root,r);
}
int main(){
    for(n=read(),i=2;i<=n;++i)j=read(),ins(j,i);
    for(i=1;i<=n;++i)a[i]=read();
    ++sz;dfs(1);++sz;build(1,sz,0);root=(1+sz)>>1;
    for(m=read();m--;){
        ch=getchar(); while (ch<'A' || ch>'Z') ch=getchar();i=read();
        if(ch=='Q')splay(L[i],root),printf("%lld\n",sum[root]-sum[c[root][1]]);
        if(ch=='C'){
            j=read();work();
            splay(x,root);splay(y,c[x][1]);z=c[y][0];c[y][0]=fa[z]=0;update(y);update(x);
            splay(L[j],root);x=root;y=find(c[x][1],1);splay(y,c[x][1]);
            c[y][0]=z;fa[z]=y;update(y);update(x);
        }
        if(ch=='F'){
            j=read();work();splay(x,root);splay(y,c[x][1]);
            add(c[y][0],j);update(y);update(x);
        }
    }
}
```

树链剖分

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
using namespace std;const int N=3e4+7,inf=1e9;char s[20];
struct data{int to,next;}e[N<<1];int
fa[N],a[N],bl[N],head[N],d[N],cnt,size[N],son[N],sum[N<<2],mx[N<<2],pos[N],sz,id[N],n,m,i,j,k;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs1(int x){
    son[x]=0;size[x]=1;
    for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
        d[e[i].to]=d[x]+1;
        fa[e[i].to]=x;
        dfs1(e[i].to);
        size[x]+=size[e[i].to];
        if(size[e[i].to]>size[son[x]])son[x]=e[i].to;
    }
}
```

```
void dfs2(int x,int chain){
	pos[x]=++sz;id[sz]=x;bl[x]=chain;
	if(son[x])dfs2(son[x],chain);
	for(int i=head[x];i;i=e[i].next)
	if(e[i].to!=fa[x]&&e[i].to!=son[x])
		dfs2(e[i].to,e[i].to);
}
void update(int rt){
	sum[rt]=sum[lson]+sum[rson];
	mx[rt]=max(mx[lson],mx[rson]);
}
void build(int rt,int l,int r){
	if(l==r){sum[rt]=mx[rt]=a[id[l]];return;}int mid=l+r>>1;
	build(lson,l,mid);build(rson,mid+1,r);update(rt);
}
void modify(int rt,int l,int r,int pos,int v){
	if(l==r){sum[rt]=mx[rt]=v;return;}int mid=l+r>>1;
	if(pos<=mid)modify(lson,l,mid,pos,v);else modify(rson,mid+1,r,pos,v);update(rt);
}
int Q1(int rt,int l,int r,int a,int b){
	if(a<=l&&r<=b)return sum[rt];int mid=l+r>>1,res=0;
	if(a<=mid)res+=Q1(lson,l,mid,a,b);if(b>mid)res+=Q1(rson,mid+1,r,a,b);return res;
}
int Q2(int rt,int l,int r,int a,int b){
	if(a<=l&&r<=b)return mx[rt];int mid=l+r>>1,res=-inf;

if(a<=mid)res=max(res,Q2(lson,l,mid,a,b));if(b>mid)res=max(res,Q2(rson,mid+1,r,a,b));return res;
}
void solve1(int x,int y){
	int res=0;
	while(bl[x]!=bl[y]){
		if(d[bl[x]]<d[bl[y]])swap(x,y);
		res+=Q1(1,1,n,pos[bl[x]],pos[x]);
		x=fa[bl[x]];
	}
	if(pos[x]>pos[y])swap(x,y);
	res+=Q1(1,1,n,pos[x],pos[y]);printf("%d\n",res);
}
void solve2(int x,int y){
	int res=-inf;
	while(bl[x]!=bl[y]){
		if(d[bl[x]]<d[bl[y]])swap(x,y);
		res=max(res,Q2(1,1,n,pos[bl[x]],pos[x]));
		x=fa[bl[x]];
```

```
    }
    if(pos[x]>pos[y])swap(x,y);
    res=max(res,Q2(1,1,n,pos[x],pos[y]));printf("%d\n",res);
}
int main(){
    for(scanf("%d",&n),i=1;i<n;++i)scanf("%d%d",&j,&k),insert(j,k);
    for(i=1;i<=n;++i)scanf("%d",a+i);dfs1(1);dfs2(1,1);build(1,1,n);
    for(scanf("%d",&m);m--;){
        scanf("%s%d%d",s,&i,&j);
        if(!strcmp(s,"QMAX"))solve2(i,j);
        if(!strcmp(s,"QSUM"))solve1(i,j);
        if(!strcmp(s,"CHANGE"))modify(1,1,n,pos[i],j);
    }
}
```

动态树

```
#include<bits/stdc++.h>
using namespace std;const int N=3e4+7,inf=1e9+7;
int fa[N],c[N][2],n,m,i,j,sz,v[N],mx[N],sum[N],q[N],top,rev[N],X[N],Y[N];char s[20];
bool isroot(int x){return c[fa[x]][0]!=x&&c[fa[x]][1]!=x;}
void update(int x){
    mx[x]=max(v[x],max(mx[c[x][0]],mx[c[x][1]]));
    sum[x]=sum[c[x][0]]+sum[c[x][1]]+v[x];
}
void rever(int x){if(!x)return;rev[x]^=1;swap(c[x][0],c[x][1]);}
void pushdown(int x){if(rev[x])rever(c[x][0]),rever(c[x][1]),rev[x]^=1;}
void rotate(int x){
    int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
    if(!isroot(y))c[z][c[z][1]==y]=x;
    fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
    c[y][l]=c[x][r];c[x][r]=y;
    update(y);update(x);
}
void splay(int x){
    q[++top]=x;for(int i=x;!isroot(i);i=fa[i])q[++top]=fa[i];
    while(top)pushdown(q[top--]);
    while(!isroot(x)){
        int y=fa[x],z=fa[y];
        if(!isroot(y)){
            if(c[y][0]==x^c[z][0]==y)rotate(x);else rotate(y);
        }rotate(x);
```

```
    }
}
void access(int x){for(int t=0;x;t=x,x=fa[x])splay(x),c[x][1]=t,update(x);}
void makeroot(int x){access(x);splay(x);rever(x);}
void link(int x,int y){makeroot(x);fa[x]=y;}
void split(int x,int y){makeroot(x);access(y);splay(y);}
int main(){
    for(scanf("%d",&n),mx[0]=-inf,i=1;i<n;++i)scanf("%d%d",X+i,Y+i);
    for(i=1;i<=n;++i)scanf("%d",v+i),mx[i]=sum[i]=v[i];
    for(i=1;i<n;++i)link(X[i],Y[i]);
    for(scanf("%d",&m);m--;){
        scanf("%s%d%d",s,&i,&j);
        if(s[0]=='C')splay(i),v[i]=j,update(i);
        else{
            split(i,j);
            if(s[1]=='M')printf("%d\n",mx[j]);else printf("%d\n",sum[j]);
        }
    }
}
```

树链上的颜色 树链剖分

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
using namespace std;const int N=1e5+7;
struct data{int to,next;}e[N<<1];int
pos[N],n,m,i,bl[N],son[N],fa[N],a[N],size[N],head[N],cnt,x,y,id[N],sz,tag[N<<2],lc[N<<2],rc[N<<2],s
um[N<<2],d[N];char ch;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dfs1(int x){
    size[x]=1;son[x]=0;
    for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
        fa[e[i].to]=x;
        d[e[i].to]=d[x]+1;
        dfs1(e[i].to);
        size[x]+=size[e[i].to];
        if(size[e[i].to]>size[son[x]])son[x]=e[i].to;
    }
}
void dfs2(int x,int chain){
```

```
    pos[x]=++sz;id[sz]=x;bl[x]=chain;
    if(son[x])dfs2(son[x],chain);
    for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]&&e[i].to!=son[x])
        dfs2(e[i].to,e[i].to);
}
int lca(int x,int y){
    for(;bl[x]!=bl[y];){
        if(d[bl[x]]<d[bl[y]])swap(x,y);x=fa[bl[x]];
    }return d[x]<d[y]?x:y;
}
void add(int rt,int val){sum[rt]=1;tag[rt]=lc[rt]=rc[rt]=val;}
void update(int rt){
    sum[rt]=sum[lson]+sum[rson]-(rc[lson]==lc[rson]);
    lc[rt]=lc[lson];rc[rt]=rc[rson];
}
void pushdown(int rt){if(tag[rt]!=-1)add(lson,tag[rt]),add(rson,tag[rt]),tag[rt]=-1;}
void build(int rt,int l,int r){
    tag[rt]=-1;if(l==r){lc[rt]=rc[rt]=a[id[l]];sum[rt]=1;return;}int mid=l+r>>1;
    build(lson,l,mid);build(rson,mid+1,r);update(rt);
}
void modify(int rt,int l,int r,int a,int b,int val){
    if(a<=l&&r<=b){add(rt,val);return;}int mid=l+r>>1;pushdown(rt);
    if(a<=mid)modify(lson,l,mid,a,b,val);if(b>mid)modify(rson,mid+1,r,a,b,val);update(rt);
}
int query(int rt,int l,int r,int a,int b){
    if(a<=l&&r<=b)return sum[rt];int mid=l+r>>1;pushdown(rt);
    if(b<=mid)return query(lson,l,mid,a,b);else if(a>mid)return query(rson,mid+1,r,a,b);
    else return query(lson,l,mid,a,b)+query(rson,mid+1,r,a,b)-(rc[lson]==lc[rson]);
}
int getc(int rt,int l,int r,int pos){
    if(l==r)return lc[rt];pushdown(rt);int mid=l+r>>1;
    if(pos<=mid)return getc(lson,l,mid,pos);else return getc(rson,mid+1,r,pos);
}
int solve(int x,int t){
    int res=0;
    while(bl[x]!=bl[t]){
        res+=query(1,1,n,pos[bl[x]],pos[x]);
        getc(1,1,n,pos[bl[x]])==getc(1,1,n,pos[fa[bl[x]]])?res--:0;
        x=fa[bl[x]];
    }res+=query(1,1,n,pos[t],pos[x]);return res;
}
void change(int x,int y,int val){
    while(bl[x]!=bl[y]){
        if(d[bl[x]]<d[bl[y]])swap(x,y);
```

```
            modify(1,1,n,pos[bl[x]],pos[x],val);
            x=fa[bl[x]];
        }
        if(d[x]>d[y])swap(x,y);
        modify(1,1,n,pos[x],pos[y],val);
}
int main(){

for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%d",a+i);for(i=1;i<n;++i)scanf("%d%d",&x,&y),inser
t(x,y);
        for(dfs1(1),dfs2(1,1),build(1,1,n);m--;){
            scanf(" %c%d%d",&ch,&x,&y);
            if(ch=='Q')i=lca(x,y),printf("%d\n",solve(x,i)+solve(y,i)-1);
            else scanf("%d",&i),change(x,y,i);
        }
}
```

树链上的颜色断 lct

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7;
int fa[N],c[N][2],v[N],lc[N],rc[N],sum[N],tag[N],rev[N],n,m,i,j,k,q[N],top;char ch;
void rever(int x){rev[x]^=1;swap(lc[x],rc[x]);swap(c[x][0],c[x][1]);}
void update(int x){
        sum[x]=sum[c[x][0]]+sum[c[x][1]]+1;
        if(rc[c[x][0]]==v[x])sum[x]--;
        if(lc[c[x][1]]==v[x])sum[x]--;
        lc[x]=rc[x]=v[x];
        if(c[x][0])lc[x]=lc[c[x][0]];
        if(c[x][1])rc[x]=rc[c[x][1]];
}
void add(int x,int val){if(!x)return;tag[x]=lc[x]=rc[x]=v[x]=val;sum[x]=1;}
void pushdown(int x){
        if(tag[x]!=-1)add(c[x][0],tag[x]),add(c[x][1],tag[x]),tag[x]=-1;
        if(rev[x])rever(c[x][0]),rever(c[x][1]),rev[x]=0;
}
bool isroot(int x){return c[fa[x]][0]!=x&&c[fa[x]][1]!=x;}
void rotate(int x){
        int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
        if(!isroot(y))c[z][c[z][1]==y]=x;
        fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
```

```
	c[y][l]=c[x][r];c[x][r]=y;
	update(y);update(x);
}
void splay(int x){
	q[++top]=x;for(int i=x;!isroot(i);i=fa[i])q[++top]=fa[i];
	while(top)pushdown(q[top--]);
	while(!isroot(x)){
		int y=fa[x],z=fa[y];
		if(!isroot(y)){
			if(c[y][0]==x^c[z][0]==y)rotate(x);else rotate(y);
		}rotate(x);
	}
}
void access(int x){for(int t=0;x;t=x,x=fa[x])splay(x),c[x][1]=t,update(x);}
void makeroot(int x){access(x);splay(x);rever(x);}
void link(int x,int y){makeroot(x);fa[x]=y;}
void split(int x,int y){makeroot(x);access(y);splay(y);}
int main(){
	lc[0]=rc[0]=v[0]=-1;memset(tag,-1,sizeof(tag));
	for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%d",v+i),lc[i]=rc[i]=v[i],sum[i]=1;
	for(i=1;i<n;++i)scanf("%d%d",&j,&k),link(j,k);
	for(;m--;){
		scanf(" %c%d%d",&ch,&i,&j);split(i,j);
		if(ch=='Q')printf("%d\n",sum[j]);else scanf("%d",&k),add(j,k);
	}
}
```

动态树完整版

```
#include<bits/stdc++.h>
using namespace std;const int N=1e4+7;
int fa[N],c[N][2],rev[N],n,m,i,j,q[N],top;char s[20];
void rever(int x){if(!x)return;rev[x]^=1;swap(c[x][0],c[x][1]);}
void pushdown(int x){if(rev[x])rever(c[x][0]),rever(c[x][1]),rev[x]=0;}
bool isroot(int x){return x!=c[fa[x]][0]&&x!=c[fa[x]][1];}
void rotate(int x){
	int y=fa[x],z=fa[y],l=c[y][1]==x,r=l^1;
	if(!isroot(y))c[z][c[z][1]==y]=x;
	fa[x]=z;fa[y]=x;fa[c[x][r]]=y;
	c[y][l]=c[x][r];c[x][r]=y;
}
void splay(int x){
```

```
	q[++top]=x;for(int i=x;!isroot(i);i=fa[i])q[++top]=fa[i];
	while(top)pushdown(q[top--]);
	while(!isroot(x)){
		int y=fa[x],z=fa[y];
		if(!isroot(y))if(c[z][0]==y^c[y][0]==x)rotate(x);else rotate(y);
		rotate(x);
	}
}
void access(int x){for(int t=0;x;t=x,x=fa[x])splay(x),c[x][1]=t;}
void makeroot(int x){access(x);splay(x);rever(x);}
void link(int x,int y){makeroot(x);fa[x]=y;}
void cut(int x,int y){makeroot(x);access(y);splay(y);c[y][0]=fa[x]=0;}
int find(int x){
	access(x);splay(x);
	while(c[x][0])x=c[x][0],pushdown(x);return x;
}
int main(){
	for(scanf("%d%d",&n,&m);m--;){
		scanf("%s%d%d",s,&i,&j);
		if(!strcmp(s,"Query"))puts(find(i)==find(j)?"Yes":"No");
		if(!strcmp(s,"Destroy"))cut(i,j);
		if(!strcmp(s,"Connect"))link(i,j);
	}
}
```

线段树合并

```
#include<bits/stdc++.h>
using namespace std;const int N=2e5+7,M=N*20;typedef long long ll;
int ls[M],rs[M],sum[M],sz,n,m,i,j;ll ans,res1,res2;
void update(int x){sum[x]=sum[ls[x]]+sum[rs[x]];}
void modify(int&x,int l,int r,int pos){
	if(!x)x=++sz;sum[x]++;if(l==r)return;int mid=l+r>>1;
	if(pos<=mid)modify(ls[x],l,mid,pos);else modify(rs[x],mid+1,r,pos);
}
int merge(int x,int y){
	if(!x||!y)return x+y;
	sum[x]=sum[x]+sum[y];
	res1+=(ll)sum[rs[x]]*sum[ls[y]];
	res2+=(ll)sum[rs[y]]*sum[ls[x]];
	ls[x]=merge(ls[x],ls[y]);
	rs[x]=merge(rs[x],rs[y]);
```

```
	update(x);return x;
}
int dfs(){
	int x,y=0,l,r;scanf("%d",&x);if(x){modify(y,1,n,x);return y;}
	l=dfs();r=dfs();res1=res2=0;y=merge(l,r);
	ans+=min(res1,res2);return y;
}
int main(){scanf("%d",&n);dfs();printf("%lld\n",ans);}
```

树上 dsu

```
#include<set>
#include<cstdio>
using namespace std;const int N=1e5+7;typedef long long ll;multiset<int>s;
struct data{int to,next;}e[N<<1];int
cnt,head[N],a[N],b[N],id[N],fa[N],L[N],R[N],son[N],size[N],sz,n,i,j,k;ll sum[N],ans[N];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void update(int x,int v){
	int c=a[x];
	s.erase(s.lower_bound(b[c]));sum[b[c]]-=c;
	b[c]+=v;
	s.insert(b[c]);sum[b[c]]+=c;
}
void dfs1(int x){
	size[x]=1;son[x]=0;L[x]=++sz;id[sz]=x;
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
		fa[e[i].to]=x;
		dfs1(e[i].to);
		size[x]+=size[e[i].to];
		if(size[e[i].to]>size[son[x]])son[x]=e[i].to;
	}R[x]=sz;
}
void dfs2(int x,int flag){
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]&&e[i].to!=son[x])dfs2(e[i].to,0);
	if(son[x])dfs2(son[x],1);
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]&&e[i].to!=son[x])
	for(int l=L[e[i].to];l<=R[e[i].to];++l)update(id[l],1);
	update(x,1);ans[x]=sum[*s.rbegin()];
	if(!flag)for(int l=L[x];l<=R[x];++l)update(id[l],-1);
}
```

```
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",a+i),s.insert(0);
	for(i=1;i<n;++i)scanf("%d%d",&j,&k),insert(j,k);
	for(dfs1(1),dfs2(1,0),i=1;i<=n;++i)printf("%lld%c",ans[i],i==n?'\n':' ');
}
```

树上启发合并

```
#include<bits/stdc++.h>
using namespace std;const int N=2e5+7;
struct data{int to,next;}e[N];int head[N],cnt,n,i,j,a[N],fa[N];multiset<int>s[N];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
int merge(int x,int y){
	if(s[x].size()<s[y].size())swap(x,y);
	for(auto&v:s[y])s[x].insert(v);
	return x;
}
void dfs(int x){
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x])dfs(e[i].to),swap(s[merge(x,e[i].to)],s[x]);
	auto p=s[x].lower_bound(a[x]);
	if(p!=s[x].end())s[x].erase(p);
	s[x].insert(a[x]);
}
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d%d",a+i,fa+i),ins(fa[i],i);
	dfs(1);printf("%d\n",s[1].size());
}
```

最小球覆盖模拟退火 **n<=60**

```
#include<cstdio>
#include<cmath>
#include<algorithm>
#define fr first
#define sc second.first
#define tr second.second
```

```
using namespace std;const int N=107;typedef pair<double,pair<double,double> >pa;const double eps=1e-7,delta=0.98;
pa a[N],now;int n,i,j;double t,ans,r;
double sqr(double x){return x*x;}
double dis(pa a,pa b){return sqrt(sqr(a.fr-b.fr)+sqr(a.sc-b.sc)+sqr(a.tr-b.tr));}
int get(pa now){int id=1;for(i=2;i<=n;++i)if(dis(now,a[id])<dis(now,a[i]))id=i;return id;}
void SA(){
	now=a[1];t=100;ans=1e60;
	while(t>eps){
		i=get(now);
		r=dis(now,a[i]);
		now.fr=now.fr+(a[i].fr-now.fr)/r*t;
		now.sc=now.sc+(a[i].sc-now.sc)/r*t;
		now.tr=now.tr+(a[i].tr-now.tr)/r*t;
		t*=delta;
		ans=min(ans,r);
	}printf("%.5f\n",ans);
}
int main(){
	for(;scanf("%d",&n),n;SA()){
		for(i=1;i<=n;++i)scanf("%lf%lf%lf",&a[i].fr,&a[i].sc,&a[i].tr);
	}
}
```

在一个区域求最小距离最大

```
#include<cstdio>
#include<cstdlib>
#include<algorithm>
#include<cmath>
#define fr first
#define sc second
using namespace std;const int N=1e3+7;const double eps=1e-3,delta=0.8;typedef pair<double,double>pa;
pa a[N],b[33],now,res;int n,m,i,j,k,T_T;double t,ans,d[N],px,py,tmp,X,Y;
double sqr(double x){return x*x;}
double dis(pa a,pa b){return sqrt(sqr(a.fr-b.fr)+sqr(a.sc-b.sc));}
double get(pa now){double tmp=1e60;for(int i=1;i<=n;++i)tmp=min(tmp,dis(now,a[i]));return tmp;}
double rnd(){return rand()/(double)RAND_MAX;}
void SA(){
	for(i=1;i<=30;++i)b[i].fr=rnd()*X,b[i].sc=rnd()*Y,d[i]=get(b[i]);t=X+Y;
```

```
    while(t>=eps){
        for(i=1;i<=30;++i){
            for(j=-1;j<=1;++j)for(k=-1;k<=1;++k){
                px=b[i].fr+t*j;
                py=b[i].sc+t*k;

if(px<=X&&px>=0&&py<=Y&&py>=0&&(tmp=get(pa(px,py)))>d[i])d[i]=tmp,b[i]=pa(px,py);
            }
        }
        t*=delta;
    }for(ans=-1,i=1;i<=30;++i)if(d[i]>ans)ans=d[i],res=b[i];
    printf("The safest point is (%.1f, %.1f).\n",res.fr,res.sc);
}
int main(){

for(scanf("%d",&T_T);T_T--;SA())for(scanf("%lf%lf%d",&X,&Y,&n),i=1;i<=n;++i)scanf("%lf%lf",&a[i].
fr,&a[i].sc);
}
```

求椭圆到原点最近的点

```
#include<bits/stdc++.h>
using namespace std;const double eps=1e-8,delta=0.99,inf=1e18;double
t,a,b,c,e,d,f,x,y,z,angle,ans,nx,ny,nz,A,B,C,judge,son1,son2;int i,j,k;
double sqr(double x){return x*x;}
double dis(double x,double y,double z){return sqrt(sqr(x)+sqr(y)+sqr(z));}
double get(double x,double y){
    A=c;B=d*y+e*x;C=a*x*x+b*y*y+f*x*y-1;
    if((judge=B*B-4*A*C)<0)return inf+10;judge=sqrt(judge);
    son1=(-B+judge)/2/A;son2=(-B-judge)/2/A;
    return dis(x,y,son1)<dis(x,y,son2)?son1:son2;
}
void SA(){
    t=1;x=0;y=0;z=sqrt(1/c);ans=z;//t 为 10 就会 wa
    while(t>=eps){
        for(i=-1;i<=1;++i)for(j=-1;j<=1;++j){
            nx=x+i*t;
            ny=y+j*t;
            if((nz=get(nx,ny))<inf&&dis(nx,ny,nz)<ans)ans=dis(nx,ny,nz),x=nx,y=ny,z=nz;
        }t*=delta;
```

```
    }printf("%.7f\n",ans);
}
int main(){
    for(;~scanf("%lf%lf%lf%lf%lf%lf",&a,&b,&c,&d,&e,&f);SA());
}
```

求 n 个点的重心模拟退火 重心就是一个点，到给个质点的距离乘权重最小的点

```
#include<bits/stdc++.h>
#define fr first
#define sc second.first
#define tr second.second
#define mp(a,b,c) make_pair(a,make_pair(b,c))
using namespace std;const int N=1e4+7;typedef pair<double,pair<double,double> >pa;const
double eps=1e-5,delta=0.8,inf=1e60;
pa a[N],now,res;int n,i,j,k;double px,py,ans,val,t;
double sqr(double x){return x*x;}
double dis(pa a,pa b){return sqrt(sqr(a.fr-b.fr)+sqr(a.sc-b.sc));}
double get(pa now){double res=0;for(int i=1;i<=n;++i)res+=dis(now,a[i])*a[i].tr;return res;}
void SA(){
    t=1e6;ans=1e60;
    while(t>=eps){
        for(i=-1;i<=1;++i)for(j=-1;j<=1;++j){
            px=now.fr+t*i;
            py=now.sc+t*j;
            if((val=get(mp(px,py,0)))<ans)ans=val,now.fr=px,now.sc=py;
        }
        t*=delta;
    }printf("%.3f %.3f\n",now.fr,now.sc);
}
int main(){
    for(scanf("%d",&n),i=1;i<=n;++i)scanf("%lf%lf%lf",&a[i].fr,&a[i].sc,&a[i].tr);SA();
}
```

把 n 个数分成 m 个数，求 m 个数方差最小 n<=20

```
#include<bits/stdc++.h>
using namespace std;const int N=27;const double eps=1e-1,delta=0.9;
int n,m,i,j,bl[N],x,y;double a[N],ans=1e60,res=1e60,sum[N],t,ave,tmp;
double sqr(double x){return x*x;}
```

```
double SA(){
	memset(sum,0,sizeof(sum));double ans=0;
	for(i=1;i<=n;++i)bl[i]=rand()%m+1,sum[bl[i]]+=a[i];t=1e4;
	for(i=1;i<=m;++i)ans+=sqr(sum[i]-ave);
	while(t>eps){
		i=rand()%n+1;
		if(t>500)j=min_element(sum+1,sum+m+1)-sum;else j=rand()%m+1;
		if(bl[i]==j)continue;
		tmp=ans;
		tmp-=sqr(sum[j]-ave);tmp-=sqr(sum[bl[i]]-ave);
		sum[bl[i]]-=a[i];sum[j]+=a[i];
		tmp+=sqr(sum[j]-ave);tmp+=sqr(sum[bl[i]]-ave);
		if(tmp<=ans||rand()%10000<=t)bl[i]=j,ans=tmp;else sum[bl[i]]+=a[i],sum[j]-=a[i];
		t*=delta;
	}return ans;
}
int main(){
	srand(983543692);
	for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%lf",a+i),ave+=a[i];ave/=m;
	for(int i=1;i<=1e4;++i)ans=min(ans,SA());printf("%.2f\n",sqrt(ans/m));
}
```

KD 树

```
#include<bits/stdc++.h>
#define L t[k].l
#define R t[k].r
using namespace std;const int N=1e5+7;typedef long long ll;
struct P{
	int d[2],mn[2],mx[2],l,r,v;ll sum;
	int operator[](int x){return d[x];}
	P(int x=0,int y=0,int z=0){d[0]=mn[0]=mx[0]=x;d[1]=mn[1]=mx[1]=y;l=r=0;v=sum=z;}
}t[N];int n,m,i,j,k,l,X,Y,H,D,root;long long ans;
bool operator<(P a,P b){return a[D]<b[D];}
void update(int k){
	t[k].sum=t[k].v+t[L].sum+t[R].sum;
	for(int i=0;i<2;++i){
		if(L)t[k].mn[i]=min(t[k].mn[i],t[L].mn[i]),t[k].mx[i]=max(t[k].mx[i],t[L].mx[i]);
		if(R)t[k].mn[i]=min(t[k].mn[i],t[R].mn[i]),t[k].mx[i]=max(t[k].mx[i],t[R].mx[i]);
	}
}
```

```
int build(int l,int r,int d){
	if(l>r)return 0;D=d;int k=l+r>>1;nth_element(t+l,t+k,t+r+1);
	L=build(l,k-1,d^1);R=build(k+1,r,d^1);
	update(k);return k;
}
bool check(ll x,ll y){return X*x+Y*y<H;}
int cal(P&a){
	int tmp=0;
	tmp+=check(a.mn[0],a.mn[1]);
	tmp+=check(a.mn[0],a.mx[1]);
	tmp+=check(a.mx[0],a.mn[1]);
	tmp+=check(a.mx[0],a.mx[1]);
	return tmp;
}
void query(int k){
	if(check(t[k][0],t[k][1]))ans+=t[k].v;
	int l=L,r=R,dl=0,dr=0;
	if(l)dl=cal(t[l]);
	if(r)dr=cal(t[r]);
	if(dl){
		if(dl==4)ans+=t[l].sum;else query(l);
	}
	if(dr){
		if(dr==4)ans+=t[r].sum;else query(r);
	}
}
int main(){

for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%d%d%d",&j,&k,&l),t[i]=P(j,k,l);root=build(1,n,0);
	for(;m--;){
		scanf("%d%d%d",&X,&Y,&H);ans=0;query(root);printf("%lld\n",ans);
	}
}
```

把序列变成递增的修改次数左偏树

```
#include<bits/stdc++.h>
using namespace std;const int N=1e6+7;
int ls[N],rs[N],size[N],v[N],a[N],d[N],sz,cnt,root[N],L[N],R[N],n,i,j;long long ans;
void update(int x){size[x]=size[ls[x]]+size[rs[x]]+1;d[x]=d[rs[x]]+1;}
int merge(int x,int y){
	if(!x||!y)return x+y;
	if(v[x]<v[y])swap(x,y);
```

```
    rs[x]=merge(rs[x],y);
    if(d[ls[x]]<d[rs[x]])swap(ls[x],rs[x]);
    update(x);return x;
}
int main(){
    for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",a+i),a[i]-=i;
    for(i=1;i<=n;++i){
        root[++cnt]=++sz;v[sz]=a[i];size[sz]=1;L[cnt]=R[cnt]=i;
        while(cnt>1&&v[root[cnt-1]]>v[root[cnt]]){
            root[cnt-1]=merge(root[cnt-1],root[cnt]);
            R[cnt-1]=R[cnt];cnt--;

while(size[root[cnt]]>(R[cnt]-L[cnt]+2)/2)root[cnt]=merge(ls[root[cnt]],rs[root[cnt]]);
        }
    }
    for(i=1;i<=cnt;++i)for(j=L[i];j<=R[i];++j)ans+=abs(v[root[i]]-a[j]);printf("%lld\n",ans);
}
```

分块求区间众数

```
#include<bits/stdc++.h>
using namespace std;const int N=4e4+7;
int n,m,a[N],i,j,k,L[N],R[N],bl[N],block,tot,ans[207][207],b[207][N],d[N],mx,id,res,l,r,tmp[N];
int main(){
    for(scanf("%d%d",&n,&m),block=sqrt(n)+1,i=1;i<=n;++i){
        scanf("%d",a+i);tot=bl[i]=i/block+1;d[i]=a[i];
        if(!L[bl[i]])L[bl[i]]=i;R[bl[i]]=i;
    }
    for(sort(d+1,d+n+1),i=1;i<=n;++i)a[i]=lower_bound(d+1,d+n+1,a[i])-d;
    for(i=1;i<=tot;++i,mx=0)for(j=i;j<=tot;ans[i][j]=id,++j){
        for(k=L[j];k<=R[j];++k){
            b[i][a[k]]++;
            if(b[i][a[k]]>mx||b[i][a[k]]==mx&&a[k]<id)mx=b[i][a[k]],id=a[k];
        }
    }
    for(;m--;printf("%d\n",res=d[id])){
        scanf("%d%d",&l,&r);l=(l+res-1)%n+1;r=(r+res-1)%n+1;if(l>r)swap(l,r);
        if(bl[l]==bl[r]||bl[l]+1==bl[r]){
            for(mx=0,id=0,i=l;i<=r;++i){
                tmp[a[i]]++;if(tmp[a[i]]==mx&&a[i]<id||tmp[a[i]]>mx)mx=tmp[a[i]],id=a[i];
            }
```

```
            for(i=l;i<=r;++i)tmp[a[i]]--;
        }else{
            id=ans[bl[l]+1][bl[r]-1];
            mx=b[bl[l]+1][id]-b[bl[r]][id];
            for(i=l;i<=R[bl[l]];++i){
                b[bl[l]+1][a[i]]++;
                if((j=b[bl[l]+1][a[i]]-b[bl[r]][a[i]])>mx||j==mx&&a[i]<id)id=a[i],mx=j;
            }
            for(i=L[bl[r]];i<=r;++i){
                b[bl[l]+1][a[i]]++;
                if((j=b[bl[l]+1][a[i]]-b[bl[r]][a[i]])>mx||j==mx&&a[i]<id)id=a[i],mx=j;
            }
            for(i=l;i<=R[bl[l]];++i)b[bl[l]+1][a[i]]--;
            for(i=L[bl[r]];i<=r;++i)b[bl[l]+1][a[i]]--;
        }
    }
}
```

区间逆序对树状数组加莫队

```
#include<bits/stdc++.h>
using namespace std;const int N=5e4+7;
struct data{int l,r,id;}e[N];int n,m,i,j,l,r,block,a[N],ans,res[N],bit[N],b[N];
bool operator<(data a,data b){return a.l/block==b.l/block?a.r<b.r:a.l<b.l;}
void add(int x,int v){while(x<=n)bit[x]+=v,x+=x&-x;}
int sum(int x,int res=0){while(x)res+=bit[x],x-=x&-x;return res;}
int main(){
    for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",a+i),b[i]=a[i];block=sqrt(n)+1;
    for(sort(b+1,b+n+1),i=1;i<=n;++i)a[i]=lower_bound(b+1,b+n+1,a[i])-b;
    for(scanf("%d",&m),i=1;i<=m;++i)scanf("%d%d",&e[i].l,&e[i].r),e[i].id=i;
    for(sort(e+1,e+m+1),l=1,r=0,i=1;i<=m;++i){
        while(r<e[i].r)++r,ans+=sum(n)-sum(a[r]),add(a[r],1);
        while(r>e[i].r)ans-=sum(n)-sum(a[r]),add(a[r],-1),--r;
        while(l<e[i].l)ans-=sum(a[l]-1),add(a[l],-1),l++;
        while(l>e[i].l)--l,ans+=sum(a[l]-1),add(a[l],1);
        res[e[i].id]=ans;
    }for(i=1;i<=m;++i)printf("%d\n",res[i]);
}
```

数位 dp　求数字出现次数

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=11;
int q[N],top,dp[N][2][N],l,r,K;
int dfs(int x,int last,int sum,int flag){
	if(x==0)return sum;if(!flag&&dp[x][last][sum]!=-1)return dp[x][last][sum];
	int mx=flag?q[x]:9,i,res=0;

for(i=0;i<=mx;++i)res+=dfs(x-1,i==0&&last==0?0:1,i==0&&last==0?sum:(sum+(i==K)),flag&&(i==m
x));
	if(!flag)dp[x][last][sum]=res;return res;
}
int cal(int x){top=0;do q[++top]=x%10,x/=10;while(x);return dfs(top,0,0,1);}
int main(){

for(;scanf("%d%d",&l,&r),l+r;)for(K=0;K<=9;++K)memset(dp,-1,sizeof(dp)),printf("%d%c",cal(max(
l,r))-cal(min(l,r)-1),K==9?'\n':' ');
}
```

数位 dp　求一个数可以整除各各数位和

```
#include<cstring>
#include<cstdio>
using namespace std;const int N=120;typedef long long ll;
ll dp[20][N][N],n;int K,q[20],top;
ll dfs(int x,int cur,int sum,int flag){
	if(x==0)return cur==0&&sum==K;
	if(!flag&&dp[x][cur][sum]!=-1)return dp[x][cur][sum];
	int mx=flag?q[x]:9,i;ll res=0;
	for(i=0;i<=mx;++i)res+=dfs(x-1,(cur*10+i)%K,sum+i,flag&&mx==i);
	if(!flag)dp[x][cur][sum]=res;return res;
}
ll cal(ll x,ll res=0){
	do q[++top]=x%10,x/=10;while(x);
	for(K=1;K<N;++K)memset(dp,-1,sizeof(dp)),res+=dfs(top,0,0,1);
	return res;
}
```

```
int main(){
	freopen("just.in","r",stdin);
	freopen("just.out","w",stdout);
	scanf("%lld",&n);printf("%lld\n",cal(n));
}
```

线段树维护区间递增序列

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
using namespace std;const int N=1e5+7;
double mx[N<<2];int ans[N<<2],n,m,i,j;
int cal(int rt,int l,int r,double val){
	if(l==r)return mx[rt]>val;int mid=l+r>>1;
	if(mx[lson]<=val)return cal(rson,mid+1,r,val);else return ans[rt]-ans[lson]+cal(lson,l,mid,val);
}
void modify(int rt,int l,int r,int pos,double val){
	if(l==r){mx[rt]=val;ans[rt]=1;return;}int mid=l+r>>1;
	if(pos<=mid)modify(lson,l,mid,pos,val);else modify(rson,mid+1,r,pos,val);
	mx[rt]=max(mx[lson],mx[rson]);
	ans[rt]=ans[lson]+cal(rson,mid+1,r,mx[lson]);
}
int main(){

for(scanf("%d%d",&n,&m);m--;)scanf("%d%d",&i,&j),modify(1,1,n,i,1.0*j/i),printf("%d\n",ans[1]);
}
```

线段树加 hall 定理　hall 定理：二分图存在 v1 到 v2 的完全匹配，当且仅当对于任意 v1 中的 K 个点，与 v2 中的 k 个点是相邻的。

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
#define fr second.first
#define sc first
#define tr second.second
using namespace std;const int N=4e5+7;long long sum=0;
int mx[N<<2],tag[N<<2],d[N],n,m,i,j,cnt,tot,flag,T_T;pair<int,pair<int,int> >s[N];
void add(int rt,int val){mx[rt]+=val;tag[rt]+=val;}
void update(int rt){mx[rt]=max(mx[lson],mx[rson]);}
void pushdown(int rt){if(tag[rt])add(lson,tag[rt]),add(rson,tag[rt]),tag[rt]=0;}
int find(int x){return lower_bound(d+1,d+tot+1,x)-d;}
void build(int rt,int l,int r){
```

```
	tag[rt]=0;if(l==r){mx[rt]=d[l]-1;return;}int mid=l+r>>1;
	build(lson,l,mid);build(rson,mid+1,r);update(rt);
}
void modify(int rt,int l,int r,int a,int b,int val){
	if(a<=l&&r<=b){add(rt,val);return;}int mid=l+r>>1;pushdown(rt);
	if(a<=mid)modify(lson,l,mid,a,b,val);if(b>mid)modify(rson,mid+1,r,a,b,val);update(rt);
}
int query(int rt,int l,int r,int a,int b){
	if(a<=l&&r<=b)return mx[rt];int mid=l+r>>1,res=-1e9;pushdown(rt);
	if(a<=mid)res=max(res,query(lson,l,mid,a,b));if(b>mid)res=max(res,query(rson,mid+1,r,a,b));
	return res;
}
int main(){
	for(mx[0]=-1e9,scanf("%d",&T_T);T_T--;tot=sum=0){

for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%d%d%d",&s[i].fr,&s[i].sc,&s[i].tr),sum+=s[i].tr;cnt
=n;
		if(sum>m){puts("No");continue;}
		for(i=1;i<=n;++i)if(s[i].fr>s[i].sc)s[i].sc+=m;else{
			s[++cnt]=s[i];
			s[i].fr+=m;s[i].sc+=m;
		}
		for(i=1;i<=cnt;++i)d[++tot]=s[i].fr,d[++tot]=s[i].sc;
		for(sort(d+1,d+tot+1),i=1;i<=cnt;++i)s[i].fr=find(s[i].fr),s[i].sc=find(s[i].sc);
		for(sort(s+1,s+cnt+1),build(1,1,tot),flag=0,i=1;i<=cnt&&!flag;++i){
			modify(1,1,tot,1,s[i].fr,s[i].tr);
			if(query(1,1,tot,find(max(0,d[s[i].sc]-m+1)),s[i].sc)>d[s[i].sc])puts("No"),flag=1;
		}if(!flag)puts("Yes");
	}
}
```

线段树求矩形面积并

```
#include<bits/stdc++.h>
#define int long long
#define lson (rt<<1)
#define rson (rt<<1|1)
using namespace std;const int N=2e5+7;
int lsum[N<<2],rsum[N<<2],mx[N<<2],sum[N<<2],n,m,k,d,i,j;
void update(int rt){
	mx[rt]=max(mx[lson],mx[rson]);
```

```
    lsum[rt]=max(lsum[lson],sum[lson]+lsum[rson]);
    rsum[rt]=max(rsum[rson],sum[rson]+rsum[lson]);
    mx[rt]=max(rsum[lson]+lsum[rson],mx[rt]);
    sum[rt]=sum[lson]+sum[rson];
}
void build(int rt,int l,int r){
    if(l==r){mx[rt]=sum[rt]=lsum[rt]=rsum[rt]=-k;return;}int mid=l+r>>1;
    build(lson,l,mid);build(rson,mid+1,r);update(rt);
}
void modify(int rt,int l,int r,int pos,int val){
    if(l==r){mx[rt]=lsum[rt]=rsum[rt]=sum[rt]=sum[rt]+val;return;}int mid=l+r>>1;
    if(pos<=mid)modify(lson,l,mid,pos,val);else modify(rson,mid+1,r,pos,val);update(rt);
}
int32_t main(){
    for(scanf("%lld%lld%lld%lld",&n,&m,&k,&d),build(1,1,n);m--;){
        scanf("%lld%lld",&i,&j);modify(1,1,n,i,j);
        puts(mx[1]<=d*k?"TAK":"NIE");
    }
}
```

BSGS 算法，中途相遇法

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;
ll y,z,mod,T,i,m,x,flag;unordered_map<ll,ll>mp;
ll pow_mod(ll x,ll n,ll mod){
    ll res=1;
    while(n){
        if(n&1)res=res*x%mod;
        x=x*x%mod;
        n>>=1;
    }
    return res;
}
int main(){
    for(;~scanf("%lld%lld%lld",&mod,&y,&z);mp.clear()){
        for(m=sqrt(mod),x=1,i=0;i<m;++i,x=x*y%mod)if(!mp.count(x))mp[x]=i;
        for(T=pow_mod(y,mod-1-m,mod),x=1,flag=i=0;i<=m+1;x=x*T%mod,++i){
            if(mp.count(x*z%mod)){
                printf("%lld\n",mp[x*z%mod]+i*m);flag=1;break;
            }
```

```
            }
            if(!flag)puts("no solution");
        }
}
```

卡特兰数列

h[0]=1 h[1]=1　h[n]=h[0]*h[n-1]+h[1]*h[n-2]+h[2]*h[n-3]...h[n-1]*h[0]

另类递推

h[n]=h[n-1]*(4*n-2)/(n+1)

h[n]=C(2*n,n)/(n+1)

h[n]=C(2*n,n)-C(2*n,n-1)

对于每一个数来说，必须进栈一次、出栈一次。我们把进栈设为状态‘1’，出栈设为状态‘0’。n 个数的所有状态对应 n 个 1 和 n 个 0 组成的 2n 位二进制数。
由于等待入栈的操作数按照 1‥n 的顺序排列、入栈的操作数 b 大于等于出栈的操作数 a(a≤b)，因此输出序列的总数目=由左而右扫描由 n 个 1 和 n 个 0 组成
的 2n 位二进制数，1 的累计数不小于 0 的累计数的方案种数。
在 2n 位二进制数中填入 n 个 1 的方案数为 c(2n,n),不填 1 的其余 n 位自动填 0。从中减去不符合要求（由左而右扫描，0 的累计数大于 1 的累计数）
的方案数即为所求。
不符合要求的数的特征是由左而右扫描时，必然在某一奇数位 2m+1 位上首先出现 m+1 个 0 的累计数和 m 个 1 的累计数，此后的 2(n-m)-1 位上有 n-m 个
1 和 n-m-1 个 0。如若把后面这 2(n-m)-1 位上的 0 和 1 互换，使之成为 n-m 个 0 和 n-m-1 个 1，结果得 1 个由 n+1 个 0 和 n-1 个 1 组成的 2n 位数，即一个不合
要求的数对应于一个由 n+1 个 0 和 n-1 个 1 组成的排列。
反过来，任何一个由 n+1 个 0 和 n-1 个 1 组成的 2n 位二进制数，由于 0 的个数多 2 个，2n 为偶数，故必在某一个奇数位上出现 0 的累计数超过 1 的累计数。
同样在后面部分 0 和 1 互换，使之成为由 n 个 0 和 n 个 1 组成的 2n 位数，即 n+1 个 0 和 n-1 个 1 组成的 2n 位数必对应一个不符合要求的数。
因而不合要求的 2n 位数与 n+1 个 0，n－1 个 1 组成的排列一一对应。
显然，不符合要求的方案数为 c(2n,n+1)。由此得出输出序列的总数目=c(2n,n)-c(2n,n-1)=c(2n,n)/(n+1)=h(n)。

```
#include<bits/stdc++.h>
#define int long long
using namespace std;const int N=21;
int ans,n,m,i,j;
int32_t main(){
    for(scanf("%lld",&n),ans=1,i=1;i<=n;++i)ans=ans*2*(2*i-1)/(i+1);
```

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

阶梯博弈，一条阶梯上有 0 1 2 3 4 5.。。。阶梯，每次可以从 i 阶梯拿任意多东西到 (i-1)阶梯上，算 nim 函数之需要算

1 3 5 ....奇数位置

```
#include<cstdio>
#include<algorithm>
using namespace std;const int N=1e3+7;
int a[N],n,m,i,j,T_T;
int main(){
    for(scanf("%d",&T_T);T_T--;){
        for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",a+i);a[++n]=0;
        for(sort(a+1,a+n+1),j=0,i=n-1;i>=1;i-=2)j^=a[i+1]-a[i]-1;
        puts(j?"Georgia will win":"Bob will win");
    }
}
```

sg 函数递推

```
#include<bits/stdc++.h>
using namespace std;const int N=1e3+7;
int sg[N],i,j,vis[N];
int main(){
    for(i=1;i<N;++i){
        for(j=1;j<=i;j<<=1)vis[sg[i-j]]=i;
        for(j=0;vis[j]==i;++j);sg[i]=j;
    }
    for(;~scanf("%d",&i);puts(sg[i]?"Kiki":"Cici"));
}
```

sg 函数

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

```
using namespace std;const int N=4e3+7;
int n,m,vis[N],sg[N],ind,kase,T_T;
int dfs(int n){
    if(sg[n]!=-1)return sg[n];//注意 sg 函数有没有赋值，没有赋值的地方，不可以直接调用 sg[n]
    if(n<m)return sg[n]=0;int i;
    for(i=1;i<=n-m+1;++i)dfs(i-1),dfs(n-(i+m-1));++ind;
    for(i=1;i<=n-m+1;++i)vis[sg[i-1]^sg[n-(i+m-1)]]=ind;
    for(i=0;;++i)if(vis[i]!=ind)return sg[n]=i;
}
int main(){
    for(scanf("%d",&T_T);T_T--;){
        scanf("%d%d",&n,&m);memset(sg,-1,sizeof(sg));printf("Case #%d: ",++kase);
        if(n<m)puts("abcdxyzk");else puts(dfs(n-m)==0?"aekdycoin":"abcdxyzk");
    }
}
```

anti nim 游戏

对于任意一个 Anti-SG 游戏， 如果我们规定当局面中所有的单一游戏的 SG 值为 0 时，游戏结束，
此时决策集合为空的选手获胜，则先手必胜当且仅当：
（1）游戏的 SG 函数不为 0 且游戏中某
个单一游戏的 SG 函数大于 1；
（2）游戏的 SG 函数为 0 且游戏中没有单一游戏的 SG 函数大于 1。

```
#include<bits/stdc++.h>
using namespace std;const int N=1e3+7;
int n,i,j,k,l,T_T,x;
int main(){
    for(scanf("%d",&T_T);T_T--;){
        for(k=l=0,scanf("%d",&n),i=1;i<=n;++i){
            scanf("%d",&x);k^=x;
            if(x>1)l=1;
        }puts(l==0&&k==0||l&&k?"John":"Brother");
    }
}
```

树上删边博弈

首先，你要知道树上删边游戏的结论
SG[x]=(SG[s1]+1) xor (SG[s2]+1) xor (SG[s3]+1)....

然后套一个环的话
我们知道，如果是一个偶环 sg 值是 0 如果是奇环 sg 值是 1

```
#include<cstdio>
#include<cstring>
const int N=1e3+7;
struct data{int to,next;}e[N];int head[N],cnt=1,q[N],ans,top,ve[N],vis[N],w[N],n,m,i,j,t;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
int dfs(int x){
	q[++top]=x;vis[x]=1;int ans=0,tmp;
	for(int i=head[x];i;i=e[i].next)if(!ve[i/2]){
		ve[i/2]=1;
		if(!vis[e[i].to])tmp=dfs(e[i].to)+1;
		else{int now=q[top--];while(now!=e[i].to)w[now]=1,now=q[top--];++top;return 1;}
		if(w[e[i].to])ans^=tmp%2;else ans^=tmp;
	}return ans;
}
int main(){
	for(;~scanf("%d",&t);ans=0){

for(;t--;memset(head,0,sizeof(head)),memset(ve,0,sizeof(ve)),memset(vis,0,sizeof(vis)),memset(
w,0,sizeof(w)),cnt=1,top=0){
			for(scanf("%d%d",&n,&m);m--;)scanf("%d%d",&i,&j),insert(i,j);ans^=dfs(1);
		}puts(ans?"Sally":"Harry");
	}
}
```

a=x1^y1 *x2^y2*x3^y3 a 的**欧拉函数**为 a*(1-1/x1)*(1-1/x2)*(1-1/x3)

```
#include<bits/stdc++.h>
using namespace std;int T_T,n,i;
int phi(int n){
	int res=n,i;
	for(i=2;i*i<=n;++i)if(n%i==0){
		res=res/i*(i-1);
```

```
            while(n%i==0)n/=i;
        }if(n!=1)res=res/n*(n-1);return res;
}
int main(){
        for(scanf("%d",&T_T);T_T--;)
            scanf("%d",&n),printf("%d\n",phi(n));
}
```

欧拉筛

```
#include<cstdio>
const int N=4e4;int phi[N],v[N],i,j,n,T_T,q[N],top;
void phi_table(){
        for(phi[1]=1,i=2;i<N;++i){
            if(!v[i])q[++top]=i,phi[i]=i-1;
            for(j=1;j<=top&&i*q[j]<N;++j){
                phi[i*q[j]]=phi[i]*(q[j]-1);v[i*q[j]]=1;
                if(i%q[j]==0){phi[i*q[j]]=phi[i]*q[j];break;}
            }
        }
}
int main(){
        for(phi_table(),scanf("%d",&T_T);T_T--;)scanf("%d",&i),printf("%d\n",phi[i]);
}
```

欧拉函数与原根个数

我们就记住吧！模 m 有原根，m=1,2,4,p,2p,p^n,p 是奇质数，当模 m 有原根时，原根的个数是 φ(φ(m))(这个是欧拉函数）

Miller rabin 加 pollard_rho 算法

```
#include<cstdio>
#include<algorithm>
using namespace std;typedef long long ll;
ll gcd(ll a,ll b){return b==0?a:gcd(b,a%b);}
```

```
ll x,mn;int T_T;
ll mul(ll a,ll b,ll mod){
	ll c=(long double)a*b/mod;
	ll ans=a*b-c*mod;
	return (ans%mod+mod)%mod;
}
ll pow_mod(ll x,ll n,ll mod){
	ll res=1;
	while(n){
		if(n&1)res=mul(x,res,mod);
		x=mul(x,x,mod);
		n>>=1;
	}
	return res;
}
bool check(ll a,ll mod,ll r,ll s){
	ll ans=pow_mod(a,r,mod),p=ans;
	for(int i=1;i<=s;++i,p=ans){
		ans=mul(ans,ans,mod);
		if(ans==1&&p!=1&&p!=mod-1)return true;
	}
	if(ans!=1)return true;
	return false;
}
bool MR(ll x){
	if(x<=1)return 0;
	if(x==2)return 1;
	if(x%2==0)return 0;
	ll r=x-1,s=0;
	while(r%2==0)r/=2,s++;
	for(int kase=20;kase--;)if(check(rand()%(x-1)+1,x,r,s))return false;
	return true;
}
ll rho(ll n,ll c){
	ll k=2,x=rand()%n,y=x,p=1;
	for(ll i=1;p==1;i++){
		x=(mul(x,x,n)+c)%n;
		p=y>x?y-x:x-y;
		p=gcd(n,p);
		if(i==k)y=x,k+=k;
	}
	return p;
}
void solve(ll n){
```

```
	if(n==1)return;
	if(MR(n)){mn=min(n,mn);return;}
	ll t=n;
	while(t==n)t=rho(n,rand()%(n-1)+1);
	solve(t);solve(n/t);
}
int main(){
	for(scanf("%d",&T_T);T_T--;){
		scanf("%lld",&x);mn=1e18;
		solve(x);
		if(mn==x)puts("Prime");
		else printf("%lld\n",mn);
	}
	return 0;
}
```

欧拉降幂公式:

A^B mod C==A^(B%mod(phi(c))+phi(c))mod C C 可以不为素数

容斥原理

```
#include<bits/stdc++.h>
#define int long long
using namespace std;const int N=2e3+7;
int a[N],n,top,b[N],i,j,y,vis[N],L,R,ans;
int gcd(int a,int b){return b==0?a:gcd(b,a%b);}
void dfs1(int x){
	if(x>R)return;
	if(x)a[++top]=x;
	dfs1(x*10+2);dfs1(x*10+9);
}
void dfs2(int x,int cnt,int y){
	if(x>n){
		if(cnt&1)ans+=R/y-(L-1)/y;
		else if(cnt)ans-=R/y-(L-1)/y;
		return;
```

```
    }
    dfs2(x+1,cnt,y);
    int tmp=y*b[x]/gcd(y,b[x]);
    if(tmp<=R)dfs2(x+1,cnt+1,tmp);
}
int32_t main(){
    cin>>L>>R;dfs1(0);sort(a+1,a+top+1);
    for(i=1;i<=top;++i)if(!vis[i])for(b[++n]=a[i],j=i;j<=top;++j)if(a[j]%a[i]==0)vis[j]=1;
    dfs2(1,0,1);cout<<ans<<endl;
}
```

容斥原理加完全背包

```
#include<cstdio>
const int N=100000+7;typedef long long ll;
int c[4],i,j,T_T,f[4],s;ll dp[N],flag,ans,res;
int main(){
    for(i=0;i<4;++i)scanf("%d",c+i);
    for(dp[0]=1,i=0;i<4;++i)for(j=0;j+c[i]<N;++j)dp[j+c[i]]+=dp[j];
    for(scanf("%d",&T_T);T_T--;ans=0){
        for(i=0;i<4;++i)scanf("%d",f+i);scanf("%d",&s);
        for(i=0;i<(1<<4);++i){
            for(j=0,flag=1,res=s;j<4;++j)if(i>>j&1)res-=(ll)(f[j]+1)*c[j],flag=-flag;
            if(res<0)res=flag=0;
            ans+=flag*dp[res];
        }
        printf("%lld\n",ans);
    }
}
```

对于任何正整数 x，其约数的个数记作 g(x)。例如 g(1)=1、g(6)=4。如果某个正整数 x 满足：g(x)>g(i) 0<i<x
，则称 x 为**反质数**。例如，整数 1，2，4，6 等都是反质数。现在给定一个数 N，你能求出不超过 N 的最大的反质数么
？

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
int p[15]={1,2,3,5,7,11,13,17,19,23,29,31},n,i,j,ans,num;
void dfs(int x,int now,int cnt,int last){
    if(x==12){
        if(cnt>num&&now>ans)ans=now,num=cnt;
```

```
            if(now<=ans&&cnt>=num)ans=now,num=cnt;
            return;
        }
        int t=1,i;
        for(i=0;i<=last;++i){
            dfs(x+1,now*t,cnt*(i+1),i);
            t*=p[x];
            if(t*now>n)break;
        }
}
int32_t main(){
        scanf("%lld",&n);dfs(1,1,1,20);printf("%lld\n",ans);
}
```

矩阵快速幂

```
#include<cstdio>
const int mod=9901;typedef long long ll;
int i,j,a,b,ans=1;ll cnt;
void mul(ll A[2][2],ll B[2][2],ll C[2][2]){
        ll tmp[2][2]={};
        for(int i=0;i<2;++i)for(int j=0;j<2;++j)for(int k=0;k<2;++k)tmp[i][j]=(tmp[i][j]+A[i][k]*B[k][j])%mod;
        for(int i=0;i<2;++i)for(int j=0;j<2;++j)C[i][j]=tmp[i][j];
}
void pow(ll A[2][2],ll n,ll C[2][2]){
        ll B[2][2]={};
        for(int i=0;i<2;++i)B[i][i]=1;
        while(n){
            if(n&1)mul(B,A,B);
            mul(A,A,A);
            n>>=1;
        }
        for(int i=0;i<2;++i)for(int j=0;j<2;++j)C[i][j]=B[i][j];
}
ll solve(ll a,ll n){
        ll A[2][2]={};
        A[0][0]=A[0][1]=1;A[1][1]=a;
        pow(A,n+1,A);
        return A[0][1];
}
```

```
void go(){
	for(ans=1,i=2;i<=a;++i)if(a%i==0){
		for(cnt=0;a%i==0;a/=i,cnt++);
		cnt*=b;
		ans=ans*solve(i,cnt)%mod;
	}
	printf("%d\n",ans);
}
int main(){
	for(;~scanf("%d%d",&a,&b);go());
}
```

扩展欧几里得

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;int x,y,d,a,b,c;
void exgcd(int a,int b,int&x,int&y){
	if(b==0)d=a,x=1,y=0;
	else exgcd(b,a%b,y,x),y-=x*(a/b);
}
int main(){
	scanf("%d%d%d",&a,&b,&c);exgcd(a,b,x,y);
	if(c%d)puts("no solution!");else printf("%lld %lld\n",(ll)x*c/d,(ll)y*c/d);
}
```

中国剩余定理

设 m1 m2 ,mn 是两两互质的整数 m=ll(i=1->i=n)mi Mi=m/mi ti 是线性同余方程方程 Mi*ti==1 (mod mi)的一个解

x=a1 (mod m1)
x=a2 (mod m2)
x=a3 (mod m3)
x=a4 (mod m4)

有整数解 x=累加 ai*Mi*ti

解线性模方程

```
#include<cstdio>
typedef long long ll;
ll a,b,c,x,y,R,M,d,A,B,flag;int n;
void exgcd(ll a,ll b,ll&x,ll&y){
	if(b==0)x=1,y=0,d=a;
	else exgcd(b,a%b,y,x),y-=x*(a/b);
}
void solve(){
	for(scanf("%lld%lld",&M,&R),flag=0,n--;n--;){
		scanf("%lld%lld",&A,&B);
		if(flag==1)continue;
		c=R-B;exgcd(M,A,x,y);
		if(c%d)flag=1;
		else{
			A/=d;
			x=(c/d*x)%A;
			R-=x*M;
			M=M*A;
			R%=M;
		}
	}
	if(!flag)printf("%lld\n",(R%M+M)%M);else puts("-1");
}
int main(){for(;~scanf("%d",&n);)solve();}
```

lucas 定理

```
#include<cstdio>
const int N=1e5+7;typedef long long ll;
ll fac[N],mod;int n,m,i,T_T;
ll pow_mod(ll x,ll n){
	ll res=1;
	while(n){
		if(n&1)res=(res*x%mod);
```

```
            x=x*x%mod;
            n>>=1;
        }
        return res;
}
ll C(int n,int m){
        if(m>n)return 0;
        return fac[n]*pow_mod(fac[n-m],mod-2)%mod*pow_mod(fac[m],mod-2)%mod;
}
ll lucas(int n,int m){
        if(m==0)return 1;
        return C(n%mod,m%mod)*lucas(n/mod,m/mod)%mod;
}
int main(){
        for(scanf("%d",&T_T);T_T--;printf("%lld\n",lucas(n+m,m))){
            for(scanf("%d%d%lld",&n,&m,&mod),fac[0]=i=1;i<=mod;++i)fac[i]=fac[i-1]*i%mod;
        }
}
```

扩展 lucas

```
#include<cstdio>
typedef long long ll;
ll n,m,mod,w[7],sum,i,res;
ll pow_mod(ll x,ll n,ll mod){
        ll res=1;
        for(;n;x=x*x%mod,n>>=1)if(n&1)res=res*x%mod;
        return res;
}
ll exgcd(ll a,ll b,ll&x,ll&y){
        if(!b)x=1,y=0;
        else exgcd(b,a%b,y,x),y-=x*(a/b);
}
ll inv(ll a,ll b){
        if(!a)return 0LL;
        ll x,y;exgcd(a,b,x,y);x=(x%b+b)%b;
        if(!x)x+=b;
        return x;
}
ll Mul(int n,int pi,int pk){
        if(n==0)return 1LL;
        ll ans=1LL;
```

```
    if(n/pk){
        for(int i=2;i<=pk;++i)if(i%pi)ans=ans*i%pk;
        ans=pow_mod(ans,n/pk,pk);
    }
    for(int i=2;i<=n%pk;++i)if(i%pi)ans=ans*i%pk;
    return ans*Mul(n/pi,pi,pk)%pk;
}
ll C(ll n,ll m,ll pi,ll pk){
    if(m>n)return 0LL;
    ll a=Mul(n,pi,pk),b=Mul(m,pi,pk),c=Mul(n-m,pi,pk);
    ll k=0LL,ans;
    for(int i=n;i;i/=pi)k+=i/pi;
    for(int i=m;i;i/=pi)k-=i/pi;
    for(int i=n-m;i;i/=pi)k-=i/pi;
    ans=a*inv(b,pk)%pk*inv(c,pk)%pk*pow_mod(pi,k,pk)%pk;
    return ans*(mod/pk)%mod*inv(mod/pk,pk)%mod;
}
ll solve(ll n,ll m){
    ll ans=0;
    for(int x=mod,i=2;i<=x;++i)if(x%i==0){
        ll pk=1LL;
        while(x%i==0)pk*=i,x/=i;
        ans=(ans+C(n,m,i,pk))%mod;
    }
    return ans;
}
int main(){
    for(scanf("%lld%lld%lld",&mod,&n,&m),i=1;i<=m;++i)scanf("%lld",w+i),sum+=w[i];
    if(sum>n)return 0*puts("Impossible");
    for(res=i=1;i<=m;++i)res=res*solve(n,w[i])%mod,n-=w[i];
    return 0*printf("%lld\n",res);
 }
```

高斯消元求方程解

```
#include<cstdio>
#include<cmath>
#include<algorithm>
using namespace std;const double eps=1e-6;
int n,i,j,k;double f[21],a[21][21],t;
double sqr(double x){return x*x;}
```

```
void gauss(){
	for(i=1;i<=n;i++){
		for(j=i;j<=n;j++)if(fabs(a[j][i])>eps)break;
		for(k=1;k<=n+1;k++)swap(a[i][k],a[j][k]);
		t=a[i][i];
		for(k=1;k<=n+1;k++)a[i][k]/=t;
		for(j=1;j<=n;j++)if(j!=i)
			for(t=a[j][i],k=1;k<=n+1;k++)
				a[j][k]-=t*a[i][k];
	}
}
int main(){
	for(scanf("%d",&n),i=1;i<=n;i++)scanf("%lf",&f[i]);
	for(i=1;i<=n;i++)
	    for(j=1;j<=n;j++){
			scanf("%lf",&t);
			a[i][j]=2*(t-f[j]);
			a[i][n+1]+=sqr(t)-sqr(f[j]);
		}
	for(gauss(),i=1;i<=n;i++)
	printf("%.3lf%c",a[i][n+1],i==n?'\n':' ');
}
```

高斯消元解决异或问题

```
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;const int N=37;
int a[N][N],n,i,j,k,l,T_T,q[N],top;
bool gauss(){
	for(top=0,i=l=1;l<=n;++l){
		for(j=i;j<=n;++j)if(a[j][l])break;if(j==n+1){q[++top]=l;continue;}
		for(k=1;k<=n+1;++k)swap(a[i][k],a[j][k]);
		for(j=1;j<=n;++j)if(a[j][l]&&j!=i)
			for(k=l;k<=n+1;++k)
				a[j][k]^=a[i][k];++i;
	}
	for(;i<=n;++i)if(a[i][n+1])return false;return true;
}
int main(){
	for(scanf("%d",&T_T);T_T--;memset(a,0,sizeof(a))){
```

```
        for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",&a[i][n+1]);
        for(i=1;i<=n;++i)scanf("%d",&j),a[i][n+1]^=j;
        for(i=1;i<=n;++i)a[i][i]=1;
        for(;scanf("%d%d",&i,&j),i+j;)a[j][i]=1;
        if(!gauss())puts("Oh,it's impossible~!!");
        else printf("%d\n",(1<<top));
    }
}
```

高斯消元求期望

```
#include<bits/stdc++.h>
#define fr first
#define sc second
using namespace std;const int N=507,M=2e5+7;typedef pair<int,int>pa;
vector<pa>e;int i,j,k,l,id,n,m,d[N];double a[N][N],t,ans,f[M];vector<int>s[N];
void gauss(){
    for(i=l=1;l<=n;++l){
        for(j=id=i;j<=n;++j)if(fabs(a[j][l])>fabs(a[id][l]))id=j;
        for(k=1;k<=n+1;++k)swap(a[i][k],a[id][k]);
        for(t=a[i][l],k=l;k<=n+1;++k)a[i][k]/=t;
        for(j=1;j<=n;++j)if(j!=i)
            for(t=a[j][l],k=l;k<=n+1;++k)
                a[j][k]-=a[i][k]*t;++i;
    }
}
int main(){
    for(scanf("%d%d",&n,&m),l=1;l<=m;++l)
    scanf("%d%d",&i,&j),e.push_back(pa(i,j)),d[i]++,d[j]++,s[i].push_back(j),s[j].push_back(i);
    for(a[1][n+1]=a[n][n+1]=1,i=1;i<=n;++i)for(a[i][i]=1,j=0;j<s[i].size();++j){
        k=s[i][j];if(k!=n)a[i][k]=-1.0/d[k];
    }
    for(gauss(),a[n][n+1]=i=0;i<m;++i)f[i]=a[e[i].fr][n+1]/d[e[i].fr]+a[e[i].sc][n+1]/d[e[i].sc];
    for(sort(f,f+m,greater<double>()),i=0;i<m;++i)ans+=(i+1)*f[i];
    printf("%.3lf\n",ans);
}
```

线性基加贪心

```
#include<bits/stdc++.h>
#define double long double
using namespace std;const int N=500+7;const double eps=1e-8;
double a[N][N],ans,t;int i,j,k,l,n,m,mn,id;
void gauss(){
	for(i=l=1;i<=n&&l<=m;++l){

for(j=i,id=-1,mn=1e9+7;j<=n;++j)if(fabs(a[j][l])>eps&&a[j][m+1]<mn)mn=a[j][m+1],id=j;if(id==-1)c
ontinue;
		for(k=1;k<=m+1;++k)swap(a[i][k],a[id][k]);ans+=a[i][m+1];
		for(j=1;j<=n;++j)if(fabs(a[j][l])>eps&&j!=i)
			for(k=1,t=a[j][l]/a[i][l];k<=m;++k)a[j][k]-=t*a[i][k];++i;
	}
}
int main(){
	for(cin>>n>>m,i=1;i<=n;++i)for(j=1;j<=m;++j)cin>>a[i][j];
	for(i=1;i<=n;++i)cin>>a[i][m+1];gauss();printf("%d %d\n",i-1,int(ans+0.5));
}
```

贪心加线性基

```
#include<bits/stdc++.h>
using namespace std;const int N=1e3+7;typedef long long ll;
ll a[N];int b[N],vis[N],id[N],i,j,k,l,ans,n;
bool cmp(int x,int y){return b[x]>b[y];}
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)scanf("%lld%d",a+i,b+i),id[i]=i;
	for(sort(id+1,id+n+1,cmp),l=0;l<=62;++l){
		for(j=1;j<=n;++j)if(!vis[id[j]]&&(a[id[j]]>>l&1))break;if(j==n+1)continue;
		vis[id[j]]=1;ans+=b[id[j]];
		for(i=1;i<=n;++i)if(!vis[id[i]]&&(a[id[i]]>>l&1))a[id[i]]^=a[id[j]];
	}printf("%d\n",ans);
}
```

逆元线性预处理

```
#include<cstdio>
const int N=1e7+7;typedef long long ll;
int inv[N],T_T,mod,fac[N],ans[N],i,j,n,m,q[N],top,prime[N];
```

```
int read(){
	int x=0;char ch=getchar();
	while(ch<'0'||ch>'9')ch=getchar();
	while(ch>='0'&&ch<='9')x=x*10+ch-'0',ch=getchar();
	return x;
}
int main(){
	for(T_T=read(),mod=read(),fac[1]=1,i=2;i<N;++i)fac[i]=(ll)fac[i-1]*i%mod;
	for(inv[1]=1,i=2;i<mod&&i<N;++i)inv[i]=ll(mod-mod/i)*inv[mod%i]%mod;
	for(ans[1]=1,i=2;i<N;++i){
		ans[i]=ans[i-1];
		if(!prime[i])ans[i]=(ll)ans[i]*(i-1)%mod*inv[i]%mod,q[++top]=i;
		for(j=1;(ll)q[j]*i<N&&j<=top;++j){
			prime[q[j]*i]=1;
			if(i%q[j]==0)break;
		}
	}
	for(;T_T--;){
		n=read();m=read();
		printf("%lld\n",(ll)fac[n]*ans[m]%mod);
	}
}
```

F(n)=累加（d|n）f(d)
f(n)=累加（d|n）U(d)*F(n/d)

模拟乌斯函数求区间[1,a] [1,b]有多少对数 gcd(a,b)==1；

```
#include<bits/stdc++.h>
using namespace std;const int N=5e4+7;typedef long long ll;
int q[N],v[N],mu[N],top,n,m,i,j,a,b,d,T_T;long long ans;
int main(){
	for(mu[1]=1,i=2;i<N;++i){
		if(!v[i])q[++top]=i,mu[i]=-1;
		for(j=1;j<=top&&i*q[j]<N;++j){
			mu[i*q[j]]=-mu[i];v[i*q[j]]=1;if(i%q[j]==0){mu[i*q[j]]=0;break;}
		}
	}
	for(i=1;i<N;++i)mu[i]+=mu[i-1];
	for(scanf("%d",&T_T);T_T--;ans=0){
		for(scanf("%d%d%d",&a,&b,&d),a/=d,b/=d,i=1;i<=min(a,b);i=j+1){
			j=min(a/(a/i),b/(b/i));
```

```
            ans+=(long long)(a/i)*(b/i)*(mu[j]-mu[i-1]);
        }printf("%lld\n",ans);
    }
}
```

莫比乌斯反演加二分 求第 K 个包含完全平方数因子的数

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;const int N=1e6+70;
int mu[N],n,m,i,j,v[N],q[N],top;ll d[N],l,r,mid,ans,K;
ll cal(ll n,ll res=0){
    for(i=2;(ll)i*i<=n;i=j+1){
        j=upper_bound(d+1,d+(int)1e6+1,n/(n/(1ll*i*i)))-d;j--;
        res+=n/(1ll*i*i)*(mu[j]-mu[i-1]);
    }return res;
}
int main(){
    for(mu[1]=1,i=2;i<N;++i){
        if(!v[i])q[++top]=i,mu[i]=1;
        for(j=1;j<=top&&i*q[j]<N;++j){
            v[i*q[j]]=1;mu[i*q[j]]=-mu[i];if(i%q[j]==0){mu[i*q[j]]=0;break;}
        }
    }for(i=1;i<N;++i)mu[i]+=mu[i-1];
    for(i=1;i<=1e6+30;++i)d[i]=((ll)i*i);
    for(scanf("%lld",&K),l=1,r=2e11;l<=r;){
        mid=l+r>>1;
        if(cal(mid)>=K)r=(ans=mid)-1;else l=mid+1;
    }printf("%lld\n",ans);
}
```

FFT 快速傅里叶变换

```
#include<bits/stdc++.h>
using namespace std;const int N=4e5;typedef complex<double>E;const double pi=acos(-1);
E a[N],b[N];int n,m,i,x,y,L,R[N];
void fft(E*x,int type){
    for(int i=0;i<n;++i)if(i<R[i])swap(x[i],x[R[i]]);
```

```
    for(int i=1;i<n;i<<=1){
        E wn(cos(pi/i),sin(type*pi/i));
        for(int p=i<<1,j=0;j<n;j+=p){
            E w(1,0);
            for(int k=0;k<i;++k,w*=wn){
                E l=x[j+k],r=w*x[j+k+i];
                x[j+k]=l+r;x[j+k+i]=l-r;
            }
        }
    }
}
int main(){
    for(scanf("%d",&n),n--;i<=n;++i)scanf("%d%d",&x,&y),a[n-i]=x,b[i]=y;
    for(m=n*2,n=1;n<=m;n<<=1)L++;
    for(i=0;i<n;++i)R[i]=(R[i>>1]>>1)|((i&1)<<(L-1));
    fft(a,1);fft(b,1);
    for(i=0;i<n;++i)a[i]=a[i]*b[i];
    fft(a,-1);
    for(i=0;i<=m/2;++i)printf("%d\n",int(a[m/2-i].real()/n+0.5));
}
```

NTT 快速数论变换

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;const ll mod=4179340454199820289LL,root=3,N=4e5+7;
int n,i,L,R[N],x,y,z,j,prime[N],t;ll a[N],b[N],c[N],d[N],inv;char ch;
ll f(ll x){return (x%mod+mod)%mod;}
ll mul(ll a,ll b){
    a%=mod;b%=mod;
    ll c=(long double)a*b/mod;
    ll ans=a*b-c*mod;
    return (ans%mod+mod)%mod;
}
ll pow_mod(ll x,ll n){
    ll res=1;
    while(n){
        if(n&1)res=mul(res,x);
        x=mul(x,x);
        n>>=1;
```

```
    }
    return res;
}
void ntt(ll*x,int type){
    for(int i=0;i<n;++i)if(i<R[i])swap(x[i],x[R[i]]);
    for(int i=1;i<n;i<<=1){
        ll wn=pow_mod(root,type==1?(mod-1)/(i<<1):mod-1-(mod-1)/(i<<1));
        for(int p=i<<1,j=0;j<n;j+=p){
            ll w=1;
            for(int k=0;k<i;++k,w=mul(w,wn)){
                ll l=x[j+k],r=mul(x[j+k+i],w);
                x[j+k]=f(l+r);x[j+k+i]=f(l-r);
            }
        }
    }
}
int main(){
    for(i=2;i<N;++i)if(!prime[i])for(j=i*2;j<N;j+=i)prime[j]=true;
    for(;scanf("%d%d%d",&x,&y,&z),x+y+z;){
        for(i=2;i<=y;++i)if(prime[i])a[i]=b[i]=c[i]=d[i]=1;
        for(;z--;){
            scanf("%d%c",&t,&ch);
            switch(ch){
                case 'S':a[t]=0;break;
                case 'H':b[t]=0;break;
                case 'C':c[t]=0;break;
                case 'D':d[t]=0;break;
            }
        }
        for(n=1,L=0;n<=y*4;n<<=1)L++;
        for(i=0;i<n;++i)R[i]=(R[i>>1]>>1)|((i&1)<<(L-1));
        ntt(a,1);ntt(b,1);ntt(c,1);ntt(d,1);
        for(i=0;i<n;++i)a[i]=mul(mul(mul(a[i],b[i]),c[i]),d[i]);
        ntt(a,-1);inv=pow_mod(n,mod-2);
        for(i=x;i<=y;++i)printf("%lld\n",mul(a[i],inv));puts("");
        for(i=0;i<n;++i)a[i]=b[i]=c[i]=d[i]=0;
    }
}
```

r*2^k+1

素数 r k 原根

3 1 1 2

5 1 2 2

17 1 4 3

97 3 5 5
193 3 6 5
257 1 8 3
7681 15 9 17
12289 3 12 11
40961 5 13 3
65537 1 16 3
786433 3 18 10
5767169 11 19 3
7340033 7 20 3
23068673 11 21 3
104857601 25 22 3
167772161 5 25 3
469762049 7 26 3
1004535809 479 21 3
2013265921 15 27 31
2281701377 17 27 3
3221225473 3 30 5
75161927681 35 31 3
77309411329 9 33 7
206158430209 3 36 22
2061584302081 15 37 7
2748779069441 5 39 3
6597069766657 3 41 5
39582418599937 9 42 5
79164837199873 9 43 5
263882790666241 15 44 7
1231453023109121 35 45 3
1337006139375617 19 46 3
3799912185593857 27 47 5
4222124650659841 15 48 19
7881299347898369 7 50 6
31525197391593473 7 52 3
180143985094819841 5 55 6
1945555039024054273 27 56 5
4179340454199820289 29 57 3

求，(1..n) (1..m)gcd(i,j)==1 的数的个数 莫比乌斯加变量代换

```
#include<bits/stdc++.h>
using namespace std;const int N=1e7+7;typedef long long ll;
```

```
int v[N],q[N],top,i,j,n,m,T_T,mu[N];ll g[N],ans;
int main(){
	for(mu[1]=1,i=2;i<N;++i){
		if(!v[i])q[++top]=i,mu[i]=-1,g[i]=1;
		for(j=1;j<=top&&i*q[j]<N;++j){
			v[i*q[j]]=1;mu[i*q[j]]=-mu[i];g[i*q[j]]=mu[i]-g[i];
			if(i%q[j]==0){mu[i*q[j]]=0;g[i*q[j]]=mu[i];break;}
		}
	}for(i=1;i<N;++i)g[i]+=g[i-1];
	for(scanf("%d",&T_T);T_T--;ans=0){
		for(scanf("%d%d",&n,&m),i=1;i<=min(n,m);i=j+1){
			j=min(n/(n/i),m/(m/i));
			ans+=(g[j]-g[i-1])*(n/i)*(m/i);
		}printf("%lld\n",ans);
	}
}
```

凸包 gramham 算法

```
#include<bits/stdc++.h>
#define sqr(x) ((x)*(x))
using namespace std;const int N=1e3+7;
struct data{
	double x,y;
	friend data operator-(data a,data b){return data{a.x-b.x,a.y-b.y};}
	friend double cross(data a,data b){return a.x*b.y-a.y*b.x;}
}p[N];int n,i,j,top;double ans;
double dis(data a,data b){return sqrt(sqr(a.x-b.x)+sqr(a.y-b.y));}
bool cmp(data a,data b){
	if(cross(a-p[0],b-p[0])==0)return dis(a,p[0])<dis(b,p[0]);
	else return cross(a-p[0],b-p[0])>0;
}
int main(){
	for(scanf("%d",&n),i=0;i<n;++i)scanf("%lf%lf",&p[i].x,&p[i].y);
	for(j=i=0;i<n;++i)if(p[i].x<p[j].x||(p[i].x==p[j].x&&p[i].y<p[j].y))j=i;swap(p[0],p[j]);
	for(sort(p+1,p+n,cmp),top=1,i=2;i<n;++i){
		while(top>=1&&cross(p[i]-p[top-1],p[top]-p[top-1])>0)top--;
		p[++top]=p[i];
	}p[++top]=p[0];
	for(i=0;i<top;++i)ans+=dis(p[i],p[i+1]);printf("%.4lf\n",ans);
}
```

平面最近点对

```
#include<bits/stdc++.h>
#define fr first
#define sc second
#define sqr(x) ((x)*(x))
#define double long double
using namespace std;const int N=5e4+7;typedef pair<double,double>pa;
const double inf=1e60;pa p[N],q[N];int n,i;
double dis(pa a,pa b){return sqrt(sqr(a.fr-b.fr)+sqr(a.sc-b.sc));}
double go(int l,int r){
	if(l==r)return inf;int mid=l+r>>1,i,j,L=l,R=mid+1,tot=l;
	double tx=p[mid].fr,d=min(go(l,mid),go(mid+1,r));
	for(;L<=mid&&R<=r;)
		if(p[L].sc<p[R].sc)q[tot++]=p[L++];else q[tot++]=p[R++];
	while(L<=mid)q[tot++]=p[L++];
	while(R<=r)q[tot++]=p[R++];
	for(i=l;i<=r;++i)p[i]=q[i];
	for(tot=0,i=l;i<=r;++i)if(fabs(p[i].fr-tx)<=d)q[++tot]=p[i];
	for(i=1;i<=tot;++i)for(j=i+1;j<=tot;++j)
	if(fabs(q[i].sc-q[j].sc)<=d)d=min(d,dis(q[i],q[j]));else break;
	return d;
}
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)scanf("%Lf%Lf",&p[i].fr,&p[i].sc);
	sort(p+1,p+n+1);printf("%.4Lf\n",go(1,n));
}
```

选 4 个点使土地面积最大旋转卡壳

```
#include<bits/stdc++.h>
#define sqr(x) ((x)*(x))
using namespace std;const int N=2e3+7;
struct data{
	double x,y;
	friend data operator-(data a,data b){return data{a.x-b.x,a.y-b.y};}
	friend double cross(data a,data b){return a.x*b.y-a.y*b.x;}
```

```
}p[N];double ans;int n,i,j,a,b,top;
double dis(data a,data b){return sqrt(sqr(a.x-b.x)+sqr(a.y-b.y));}
bool cmp(data a,data b){
	if(cross(a-p[0],b-p[0])==0)return dis(a,p[0])<dis(b,p[0]);
	else return cross(a-p[0],b-p[0])>0;
}
int main(){
	for(scanf("%d",&n),i=0;i<n;++i)scanf("%lf%lf",&p[i].x,&p[i].y);
	for(i=j=0;i<n;++i)if(p[i].x<p[j].x||(p[i].x==p[j].x&&p[i].y<p[j].y))j=i;swap(p[0],p[j]);
	for(sort(p+1,p+n,cmp),top=1,i=2;i<n;++i){
		while(top>=1&&cross(p[i]-p[top-1],p[top]-p[top-1])>=0)top--;
		p[++top]=p[i];
	}
	for(++top,i=0;i<top;++i)for(a=i,j=b=i+1;j<top;++j){

while((a+1)%top!=j&&fabs(cross(p[j]-p[i],p[(a+1)%top]-p[i]))>fabs(cross(p[j]-p[i],p[a]-p[i])))a=(a+1)
%top;

while((b+1)%top!=i&&fabs(cross(p[i]-p[j],p[(b+1)%top]-p[j]))>fabs(cross(p[i]-p[j],p[b]-p[j])))b=(b+
1)%top;
		ans=max(ans,fabs(cross(p[a]-p[i],p[j]-p[i]))+fabs(cross(p[b]-p[j],p[i]-p[j])));
	}printf("%.3lf\n",ans/2);
}
```

舞蹈链精确覆盖

```
#include<bits/stdc++.h>
using namespace std;const int M=100010,N=1010;
int n,m,num,i,j;
struct DLX{
	int U[M],D[M],R[M],L[M],Row[M],Col[M];
	int H[N],S[N],ansd,ans[N],n,m,size;
	void init(int _n,int _m){
		n = _n;
		m = _m;
		for(int i = 0;i <= m;i++){
			S[i] = 0;
			U[i] = D[i] = i;
			L[i] = i-1;
			R[i] = i+1;
		}
		R[m] = 0; L[0] = m;
```

```
    size = m;
    for(int i = 1;i <= n;i++)H[i] = -1;
}
void Link(int r,int c){
    ++S[Col[++size]=c];
    Row[size] = r;
    D[size] = D[c];
    U[D[c]] = size;
    U[size] = c;
    D[c] = size;
    if(H[r] < 0)H[r] = L[size] = R[size] = size;
    else{
        R[size] = R[H[r]];
        L[R[H[r]]] = size;
        L[size] = H[r];
        R[H[r]] = size;
    }
}
void remove(int c){
    L[R[c]] = L[c]; R[L[c]] = R[c];
    for(int i = U[c];i != c;i = U[i])
        for(int j = L[i];j != i;j = L[j]){
            U[D[j]] = U[j];
            D[U[j]] = D[j];
            --S[Col[j]];
        }
}
void resume(int c){
    for(int i = U[c];i != c;i = U[i])
        for(int j = L[i];j != i;j = L[j])
            ++S[Col[U[D[j]]=D[U[j]]=j]];
    L[R[c]] = R[L[c]] = c;
}
bool Dance(int d){      //d 为递归深度
    if(L[0] == 0){ansd=d;return true;}
    int c = L[0];
    for(int i = L[0];i != 0;i = L[i])
        if(S[i] < S[c])
            c = i;
    remove(c);
    for(int i = U[c];i != c;i = U[i]){
        ans[d] = Row[i];
        for(int j = L[i]; j != i;j = L[j])remove(Col[j]);
        if(Dance(d+1))return true;
```

```
            for(int j = R[i]; j != i;j = R[j])resume(Col[j]);
        }
        resume(c);
        return false;
    }
}g;
int main(){
    while(scanf("%d%d",&n,&m) == 2){
        for(g.init(n,m),i=1;i<=n;i++)
            for(scanf("%d",&num);num--;)scanf("%d",&j),g.Link(i,j);
        if(!g.Dance(0))printf("NO\n");
        else{
            for(printf("%d",g.ansd),i=0;i<g.ansd;i++)
                printf(" %d",g.ans[i]);puts("");
        }
    }
}
```

舞蹈链重复覆盖

```
#include<cstdio>
#include<cstring>
const int N=400,M=500005;
int n,m,i,j,k,l,x,n1,m1,id[30][30],ind;
struct DLX{
    int U[M],D[M],L[M],R[M],Row[M],Col[M];
    int H[N],S[N],ans,vis[N],n,m,size;
    void init(int _n,int _m){
        n=_n;
        m=_m;
        for(int i=0;i<=m;++i){
            S[i]=0;
            U[i]=D[i]=i;
            L[i]=i-1;
            R[i]=i+1;
        }
        R[m]=0;L[0]=m;
        size=m;ans=1e9;
        for(int i=1;i<=n;++i)H[i]=-1;
    }
    void Link(int r,int c){
```

```
        ++S[Col[++size]=c];
        Row[size]=r;
        D[size]=D[c];
        U[D[c]]=size;
        U[size]=c;
        D[c]=size;
        if(H[r]<0)H[r]=L[size]=R[size]=size;
        else{
            R[size]=R[H[r]];
            L[R[H[r]]]=size;
            L[size]=H[r];
            R[H[r]]=size;
        }
    }
    void remove(int c){
        for(int i=U[c];i!=c;i=U[i])
            L[R[i]]=L[i],R[L[i]]=R[i];
    }
    void resume(int c){
        for(int i=U[c];i!=c;i=U[i])
            L[R[i]]=R[L[i]]=i;
    }
    int f(int res=0){
        for(int i=L[0];i!=0;i=L[i])
            vis[i]=1;
```

目录

```
        for(int i=L[0];i!=0;i=L[i])
            if(vis[i]){
                vis[i]=0;res++;
                for(int j=U[i];j!=i;j=U[j])
                    for(int k=L[j];k!=j;k=L[k])
                        vis[Col[k]]=0;
            }
        return res;
    }
    void Dance(int d){
        if(d+f()>=ans)return;
        if(L[0]==0){if(d<ans)ans=d;return;}
        int c=L[0];
        for(int i=L[0];i!=0;i=L[i])
            if(S[i]<S[c])
                c=i;
        for(int i=U[c];i!=c;i=U[i]){
```

```
                remove(i);
                for(int j=L[i];j!=i;j=L[j])remove(j);
                Dance(d+1);
                for(int j=R[i];j!=i;j=R[j])resume(j);
                resume(i);
            }
        }
}g;
int main(){
    for(;scanf("%d%d",&n,&m)!=EOF;g.Dance(0),printf("%d\n",g.ans),memset(id,0,sizeof(id)),in
d=0){
            for(i=1;i<=n;++i)for(j=1;j<=m;++j){
                scanf("%d",&x);
                if(x)id[i][j]=++ind;
            }
            for(scanf("%d%d",&n1,&m1),g.init(n*m,ind),ind=i=1;i<=n-n1+1;++i)
            for(j=1;j<=m-m1+1;++j,++ind)for(k=i;k<i+n1;++k)for(l=j;l<j+m1;++l)
                if(id[k][l])g.Link(ind,id[k][l]);
    }
}
```

三分法

```
#include<bits/stdc++.h>
using namespace std;const int N=1e4+7;typedef long long ll;
int l,r,mid1,mid2,i,n,x[N];ll t1,t2,ans=1e18;
ll cal(ll y,ll res=0){for(i=1;i<=n;++i)res+=abs(x[i]-y);return res;}
int main(){
    for(scanf("%d",&n),l=2e9,r=-2e9,i=1;i<=n;++i)scanf("%d",x+i),l=min(l,x[i]),r=max(r,x[i]);
    for(ans=min({cal(l),cal(r),cal(l+r>>1)});r-l>2;){
        mid1=l+(r-l)/3;
        mid2=r-(r-l)/3;
        if((t1=cal(mid1))<(t2=cal(mid2)))r=mid2;else l=mid1;
        ans=min({ans,t1,t2});
    }printf("%lld\n",ans);
}
```

杜教筛求莫比乌斯函数前缀和

空间大小为 pow(n,2/3) 时间复杂度为 pow(n,2/3)*log(n)

C(n)=累加（d|n）A(d)B(n/d)

则有
SA(n)=SC(n)-累加（i=2..i=n）B(i)*SA([n/i]);

1 ==累加（d|n）mu[d]　n 为任意数

1^2 + 2^2 + 3^2 + 4^2 + 5^2。。。。n^2=n*(n+1)*(2*n+1)/6

```
#include<bits/stdc++.h>
using namespace std;const int N=3981071;typedef long long ll;
int q[N],v[N],top,i,j;ll mu[N],l,r;unordered_map<ll,int>H;
ll DJ_shai(ll n){
	if(n<N)return mu[n];else if(H.count(n))return H[n];
	ll res=0,i,j;
	for(i=2;i<=n;i=j+1){
		j=n/(n/i);
		res+=(j-i+1)*DJ_shai(n/i);
	}return H[n]=1-res;
}
int main(){
	for(mu[1]=1,i=2;i<N;++i){
		if(!v[i])q[++top]=i,mu[i]=-1;
		for(j=1;j<=top&&i*q[j]<N;++j){
			v[i*q[j]]=1;mu[i*q[j]]=-mu[i];
			if(i%q[j]==0){mu[i*q[j]]=0;break;}
		}
	}for(i=1;i<N;++i)mu[i]+=mu[i-1];
	scanf("%lld%lld",&l,&r);printf("%lld\n",DJ_shai(r)-DJ_shai(l-1));
}
```

Sphi(n)=n*(n+1)/2-累加（i=2..i=n）Sphi([n/i]);

杜教筛求欧拉函数和

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;const int N=3981071,mod=1e9+7,inv=500000004;
int q[N],v[N],top,i,j;ll phi[N],n;unordered_map<ll,int>H;
ll DJ_shai(ll n){
	if(n<N)return phi[n];else if(H.count(n))return H[n];
	ll res=0,i,j;
	for(i=2;i<=n;i=j+1){
		j=n/(n/i);
		res+=(j-i+1)*DJ_shai(n/i);
	}return ((H[n]=n%mod*((n+1)%mod)%mod*inv%mod-res%mod)%mod+mod)%mod;
}
int main(){
	for(phi[1]=1,i=2;i<N;++i){
		if(!v[i])q[++top]=i,phi[i]=i-1;
		for(j=1;j<=top&&i*q[j]<N;++j){
			v[i*q[j]]=1;phi[i*q[j]]=phi[i]*phi[q[j]];
			if(i%q[j]==0){phi[i*q[j]]=phi[i]*q[j];break;}
		}
	}for(i=1;i<N;++i)phi[i]=(phi[i]+phi[i-1])%mod;
	scanf("%lld",&n);printf("%lld\n",DJ_shai(n));
}
```

FWT

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7,mod=(1e9+7);typedef long long ll;
int n,i,j,m,prime[N],q[N],top,t;ll a[N],inv;
void init(){
	for(i=2;i<N;++i){
		if(!prime[i])q[++top]=i;
		for(j=1;i*q[j]<N&&j<=top;++j){
			prime[i*q[j]]=1;
			if(i%q[j]==0)break;
		}
	}
```

```
}
ll pow_mod(ll x,ll n){
	ll res=1;
	while(n){
		if(n&1)res=res*x%mod;
		x=x*x%mod;
		n>>=1;
	}
	return res;
}
void fwt(ll *x,int n,int flag){
	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){
				ll l=x[j+k],r=x[j+k+i];
				if(flag==1)x[j+k]=(l+r)%mod,x[j+k+i]=(l-r+mod)%mod;
				else x[j+k]=(l+r)*inv%mod,x[j+k+i]=(l-r+mod)%mod*inv%mod;
			}
		}
	}
}
int main(){
	for(init(),inv=pow_mod(2,mod-2);~scanf("%d%d",&t,&m);memset(a,0,sizeof(a))){
		for(i=1;i<=top&&q[i]<=m;++i)a[q[i]]=1;
		for(n=1;n<=m;n<<=1);fwt(a,n,1);
		for(i=0;i<n;++i)a[i]=pow_mod(a[i],t);
		fwt(a,n,-1);printf("%lld\n",a[0]);
	}
}

/*
void FWT_or(int *a,int opt)
{
	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)
				if(opt==1)a[i+j+k]=(a[j+k]+a[i+j+k])%MOD;
				else a[i+j+k]=(a[i+j+k]+MOD-a[j+k])%MOD;
}
void FWT_and(int *a,int opt)
{
	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)
```

```
                    if(opt==1)a[j+k]=(a[j+k]+a[i+j+k])%MOD;
                    else a[j+k]=(a[j+k]+MOD-a[i+j+k])%MOD;
}
void FWT_xor(int *a,int opt)
{
    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];
                a[j+k]=(X+Y)%MOD;a[i+j+k]=(X+MOD-Y)%MOD;
                if(opt==-1)a[j+k]=1ll*a[j+k]*inv2%MOD,a[i+j+k]=1ll*a[i+j+k]*inv2%MOD;
            }
}

*/
```

Polya 定理

```
#include<cstdio>
#include<cstring>
const int N=11,M=1e5+7,mod=9973;
int n,m,Q,A[N][N],v[M],q[M],top,T_T,i,j,ans;
int pow_mod(int x,int n){
    int res=1;
    while(n){
        if(n&1)res=res*x%mod;
        x=x*x%mod;
        n>>=1;
    }return res;
}
void mul(int A[N][N],int B[N][N],int C[N][N]){
    int tmp[N][N]={};
    for(int i=1;i<=m;++i)for(int j=1;j<=m;++j)for(int k=1;k<=m;++k)
    tmp[i][k]=(tmp[i][k]+A[i][j]*B[j][k])%mod;
    for(int i=1;i<=m;++i)for(int j=1;j<=m;++j)C[i][j]=tmp[i][j];
}
void pow_mod(int A[N][N],int n,int C[N][N]){
    int res[N][N]={};
    for(int i=1;i<=m;++i)res[i][i]=1;
    while(n){
        if(n&1)mul(res,A,res);
        mul(A,A,A);
```

```
            n>>=1;
        }
        for(int i=1;i<=m;++i)for(int j=1;j<=m;++j)C[i][j]=res[i][j];
}
int phi(int x){
        int res=x;
        for(int i=1;q[i]*q[i]<=x;++i)if(x%q[i]==0){
            res=res/q[i]*(q[i]-1);while(x%q[i]==0)x/=q[i];
        }
        if(x!=1)res=res/x*(x-1);
        return res;
}
int num(int n){
        int a[N][N];memcpy(a,A,sizeof(A));int res=0;
        pow_mod(a,n,a);for(int i=1;i<=m;++i)res+=a[i][i],res%=mod;
        return res;
}
void solve(int x){ans+=phi(x)%mod*num(n/x)%mod;ans%=mod;}
int main(){
        for(i=2;i<M;++i){
            if(!v[i])q[++top]=i;
            for(j=1;j<=top&&i*q[j]<M;++j){
                v[i*q[j]]=1;if(i%q[j]==0)break;
            }
        }
        for(scanf("%d",&T_T);T_T--;ans=0){
            for(scanf("%d%d%d",&n,&m,&Q),i=1;i<=m;++i)for(j=1;j<=m;++j)A[i][j]=1;
            for(;Q--;)scanf("%d%d",&i,&j),A[i][j]=A[j][i]=0;
            for(i=1;i*i<=n;++i)if(n%i==0){
                solve(i);
                if(i!=n/i)solve(n/i);
            }printf("%d\n",ans*pow_mod(n%mod,mod-2)%mod);
        }
}
```

拉格朗日插值

```
#include<bits/stdc++.h>
using namespace std;const int N=2e3+7,mod=998244353;typedef long long ll;
int n,k,x[N],y[N],i,j;ll fz,fm,sum;
ll pow_mod(ll x,ll n){
```

```
    ll res=1;
    while(n){
        if(n&1)res=res*x%mod;
        x=x*x%mod;
        n>>=1;
    }return res;
}
void mul(ll&x,ll v){v=(v%mod+mod)%mod;x=x*v%mod;}
int main(){
    for(scanf("%d%d",&n,&k),i=1;i<=n;++i)scanf("%d%d",x+i,y+i);
    for(i=1;i<=n;++i){
        for(fz=y[i],fm=1,j=1;j<=n;++j)if(i!=j){
            mul(fz,(k-x[j]));
            mul(fm,(x[i]-x[j]));
        }
        fm=pow_mod(fm,mod-2);
        sum=(sum+fz*fm)%mod;
    }printf("%lld\n",sum);
}
/*
3 100
1 4
2 9
3 16
*/
```

第二类 Stirling 数加树上 dp 加公式

```
#include<bits/stdc++.h>
using namespace std;const int N=5e4+7,M=5e2+7,mod=10007;
struct data{int to,next;}e[N<<1];int
head[N],T_T,cnt,fa[N],d[N][M],u[N][M],S[M][M],fac[M],n,m,i,j,k,ans;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void add(int&x,int v){(x+=v%mod+mod)%=mod;}
void down(int x){
    d[x][0]=1;
    for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
        fa[e[i].to]=x;
        down(e[i].to);
        add(d[x][0],d[e[i].to][0]);
        for(int j=1;j<=m;++j)add(d[x][j],d[e[i].to][j-1]+d[e[i].to][j]);
```

```
	}
}
void up(int x){
	if(fa[x]){
		u[x][0]=n-d[x][0];
		for(int j=1;j<=m;++j){
			u[x][j]=(u[fa[x]][j-1]+u[fa[x]][j]+d[fa[x]][j-1]+d[fa[x]][j]-d[x][j]-2*d[x][j-1]);
			if(j>1)add(u[x][j],-d[x][j-2]);
		}
	}
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x])up(e[i].to);
}
int main(){
	for(S[0][0]=1,i=1;i<M;++i)for(j=1;j<=i;++j)S[i][j]=(S[i-1][j-1]+S[i-1][j]*j)%mod;
	for(fac[0]=1,i=1;i<M;++i)fac[i]=fac[i-1]*i%mod;
	for(scanf("%d",&T_T);T_T--;memset(head,0,sizeof(head)),cnt=0,memset(d,0,sizeof(d)),mem
set(u,0,sizeof(u))){
		for(scanf("%d%d",&n,&m),i=1;i<n;++i)scanf("%d%d",&j,&k),insert(j,k);
		for(down(1),up(1),i=1;i<=n;++i){
			for(ans=0,j=1;j<=m;++j)add(ans,S[m][j]*fac[j]%mod*(d[i][j]+u[i][j])%mod);
			printf("%d\n",ans);
		}
	}
}
/*
xk=∑(i=1->k)Stirling2(k,i)×i!×C(x,i)
设 f[i][j]=∑(k=1->n)C(dist(i,k),j)。

则可以利用 C(i,j)=C(i-1,j-1)+C(i-1,j)，通过树形 DP 求出 f。
*/
```

第一类 stirling 数

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;const int N=22;
ll fac[N],f[N][N],ans;int n,m,i,j,T_T;
int main(){
	for(fac[0]=1,i=1;i<N;++i)fac[i]=fac[i-1]*i;
	for(f[0][0]=1,i=1;i<N;++i)for(j=1;j<=i;++j)f[i][j]=f[i-1][j-1]+f[i-1][j]*(i-1);
	for(scanf("%d",&T_T);T_T--;){
```

```
        for(scanf("%d%d",&n,&m),ans=0,i=1;i<=m;++i)ans+=f[n][i]-f[n-1][i-1];
        printf("%.4f\n",1.0*ans/fac[n]);
    }
}
```

斯坦纳树

```
#include<bits/stdc++.h>
namespace FIFO{
    char ch,B[1<<20],*S=B,*T=B;
    #define getc() (S==T&&(T=(S=B)+fread(B,1,1<<20,stdin),S==T)?0:*S++)
    #define isd(c) (c>='0'&&c<='9')
    int aa,bb;int F(){
        while(ch=getc(),!isd(ch)&&ch!='-');ch=='-'?(aa=bb=0):(aa=ch-'0',bb=1);
        while(ch=getc(),isd(ch))aa=aa*10+ch-'0';return bb?aa:-aa;
    }
}
#define gi FIFO::F()
using namespace std;const int N=1e3+7,M=(1<<10)+30,inf=0x3f3f3f3f;
struct data{int to,next,v;}e[N*10];int
head[N],cnt,inq[N],g[N],dp[N][M],id[11],i,j,k,x,col[11],sum[11],tmp[11],t,w,n,m,K,q[N];
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,w);}
void spfa(int S){
    while(t!=w){
        x=q[t++];inq[x]=0;if(t==N)t=0;
        for(int i=head[x];i;i=e[i].next)if(dp[x][S]+e[i].v<dp[e[i].to][S]){
            dp[e[i].to][S]=dp[x][S]+e[i].v;
            if(!inq[e[i].to]){
                q[w++]=e[i].to;
                inq[e[i].to]=1;
                if(w==N)w=0;
            }
        }
    }
}
bool check(int S){
    memset(tmp,0,sizeof(tmp));
    for(int i=0;i<K;++i)if(S>>i&1)tmp[col[i]]++;
    for(int i=1;i<=K;++i)if(tmp[i]&&tmp[i]!=sum[i])return false;
    return true;
}
```

```
int main(){
	for(n=gi,m=gi,K=gi;m--;)i=gi,j=gi,k=gi,insert(i,j,k);
	memset(dp,inf,sizeof(dp));memset(g,inf,sizeof(g));
	for(i=0;i<K;++i)col[i]=gi,id[i]=gi,sum[col[i]]++,dp[id[i]][(1<<i)]=0;
	for(k=1;k<(1<<K);++k){
		for(t=w=0,i=1;i<=n;++i){
			for(j=k;j;j=(j-1)&k)dp[i][k]=min(dp[i][k],dp[i][j]+dp[i][k^j]);
			if(dp[i][k]!=inf)q[w++]=i,inq[i]=1;
		}
		spfa(k);
	}
	for(i=1;i<(1<<K);++i)for(j=1;j<=n;++j)g[i]=min(g[i],dp[j][i]);
	for(i=1;i<(1<<K);++i)if(check(i))for(j=i;j;j=(j-1)&i)if(check(j))
	g[i]=min(g[i],g[i^j]+g[j]);
	printf("%d\n",g[(1<<K)-1]);
}
```

最小树形图

```
#include<bits/stdc++.h>
using namespace std;const int N=1e5+7,inf=1e9+7;
int u[N],v[N],w[N],n,m,R,i,j,x,t,cnt,mn[N],vis[N],id[N],fa[N];
int ZhuLiu(){
	int ans=0;
	while(true){
		for(i=1;i<=n;++i)id[i]=vis[i]=0,mn[i]=inf;cnt=mn[R]=0;
		for(i=1;i<=m;++i)if(u[i]!=v[i]&&w[i]<mn[v[i]])mn[v[i]]=w[i],fa[v[i]]=u[i];
		for(i=1;i<=n;++i){
			if(mn[i]==inf)return -1;ans+=mn[i];
			for(x=i;(!id[x])&&x!=R&&vis[x]!=i;x=fa[x])vis[x]=i;
			if(x!=R&&(!id[x])){
				id[x]=++cnt;for(t=fa[x];t!=x;t=fa[t])id[t]=cnt;
			}
		}
		if(!cnt)return ans;
		for(i=1;i<=n;++i)if(!id[i])id[i]=++cnt;
		for(i=1;i<=m;++i){
			t=mn[v[i]];
			if((u[i]=id[u[i]])!=(v[i]=id[v[i]]))w[i]-=t;
		}n=cnt;R=id[R];
	}return ans;
```

```
}
int main(){
	for(scanf("%d%d%d",&n,&m,&R),i=1;i<=m;++i)scanf("%d%d%d",u+i,v+i,w+i);
	printf("%d\n",ZhuLiu());
}
/*
4 6 1
1 2 3
1 3 1
4 1 2
4 2 2
3 2 1
3 4 1
*/
```

仙人掌图

```
#include<bits/stdc++.h>
using namespace std;const int N=5e4+7,M=1e6+7;
struct data{int to,next;}e[M<<1];int
n,m,i,j,head[N],f[N],ans,tot,x,y,cnt,q[N*2],t,w,a[N*2],ind,d[N],fa[N],dfn[N],low[N];
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void insert(int u,int v){ins(u,v);ins(v,u);}
void dp(int root,int x){
	for(tot=d[x]-d[root]+1,i=x;i!=root;i=fa[i])a[tot--]=f[i];
	a[1]=f[root];
	for(tot=d[x]-d[root]+1,i=1;i<=tot;++i)a[i+tot]=a[i];
	for(q[0]=1,t=0,w=1,i=2;i<=tot*2;++i){
		while(t<w&&i-q[t]>tot/2)t++;
		ans=max(ans,a[i]+i+a[q[t]]-q[t]);
		while(t<w&&a[q[w-1]]-q[w-1]<=a[i]-i)w--;
		q[w++]=i;
	}
	for(i=2;i<=tot;++i)
		f[root]=max(f[root],a[i]+min(i-1,tot-i+1));
}
void dfs(int x){
	dfn[x]=low[x]=++ind;
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
		if(!dfn[e[i].to]){
			fa[e[i].to]=x;
			d[e[i].to]=d[x]+1;
```

```
                dfs(e[i].to);
                low[x]=min(low[x],low[e[i].to]);
            }else low[x]=min(low[x],dfn[e[i].to]);
            if(dfn[x]<low[e[i].to]){
                ans=max(ans,f[x]+f[e[i].to]+1);
                f[x]=max(f[x],f[e[i].to]+1);
            }
        }
        for(int i=head[x];i;i=e[i].next)
            if(fa[e[i].to]!=x&&dfn[x]<dfn[e[i].to])
                dp(x,e[i].to);
}
int main(){
        for(scanf("%d%d",&n,&m);m--;)for(scanf("%d%d",&j,&x),j--;j--;x=y)scanf("%d",&y),insert(x,y)
;
        dfs(1);printf("%d\n",ans);
}
```

圆方树

```
#include<bits/stdc++.h>
using namespace std;const int N=4e4+7;
struct data{int to,next,v;}e[N<<3];int cnt;
struct graph{
        int head[N];
        void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
        void insert(int u,int v,int w){ins(u,v,w);ins(v,u,w);}
}g1,g2;int ans,n,m,q,u,v,w,i,j,fa[N],sum[N],b[N],sz,f[N][20],h[N],d[N],dfn[N],low[N],ind;
void solve(int u,int v,int w){
        int tot=w;
        for(int i=v;i!=fa[u];i=fa[i]){
            sum[i]=tot;
            tot+=b[i];
        }
        ++sz;sum[sz]=sum[u];sum[u]=0;
        for(int i=v;i!=fa[u];i=fa[i])g2.insert(sz,i,min(sum[sz]-sum[i],sum[i]));
}
void dfs(int x){
        dfn[x]=low[x]=++ind;
        for(int i=g1.head[x];i;i=e[i].next)if(e[i].to!=fa[x]){
            if(!dfn[e[i].to]){
```

```
                fa[e[i].to]=x;
                b[e[i].to]=e[i].v;
                dfs(e[i].to);
                low[x]=min(low[x],low[e[i].to]);
            }else low[x]=min(low[x],dfn[e[i].to]);
            if(dfn[x]<low[e[i].to])g2.insert(x,e[i].to,e[i].v);
        }
        for(int i=g1.head[x];i;i=e[i].next)
            if(fa[e[i].to]!=x&&dfn[x]<dfn[e[i].to])
                solve(x,e[i].to,e[i].v);
}
void dfs1(int x){
        for(int i=1;(1<<i)<=h[x];++i)f[x][i]=f[f[x][i-1]][i-1];
        for(int i=g2.head[x];i;i=e[i].next)if(e[i].to!=f[x][0])
            h[e[i].to]=h[x]+1,d[e[i].to]=d[x]+e[i].v,f[e[i].to][0]=x,dfs1(e[i].to);
}
int go(int x,int t){for(int i=19;i>=0;--i)if(t>>i&1)x=f[x][i];return x;}
int lca(int x,int y){
        if(h[x]<h[y])swap(x,y);x=go(x,h[x]-h[y]);
        for(int i=19;i>=0;--i)if(f[x][i]!=f[y][i])x=f[x][i],y=f[y][i];
        return x==y?x:f[x][0];
}
int main(){
        for(scanf("%d%d%d",&n,&m,&q);m--;)scanf("%d%d%d",&u,&v,&w),g1.insert(u,v,w);sz=n;
        for(dfs(1),dfs1(1);q--;){
            scanf("%d%d",&u,&v);w=lca(u,v);
            if(w<=n)ans=d[u]+d[v]-2*d[w];
            else {
                i=go(u,h[u]-h[w]-1);
                j=go(v,h[v]-h[w]-1);
                ans=d[u]+d[v]-d[i]-d[j];
                if(sum[i]>sum[j])swap(i,j);
                ans+=min(sum[j]-sum[i],sum[w]+sum[i]-sum[j]);
            }printf("%d\n",ans);
        }
}
```

多项式求逆

```
#include<bits/stdc++.h>
using namespace std;const int mod=1005060097,root=5,N=600003;typedef long long ll;
```

```
ll a[N],b[N],c[N];int n,m,t,i,L,mx,x,R[N];
ll pow_mod(ll x,ll n){
	ll res=1;
	while(n){
		if(n&1)res=res*x%mod;
		x=x*x%mod;
		n>>=1;
	}
	return res;
}
void ntt(ll*x,int n,int L,int type){
	for(int i=0;i<n;++i)R[i]=(R[i>>1]>>1)|((i&1)<<(L-1));
	for(int i=0;i<n;++i)if(i<R[i])swap(x[i],x[R[i]]);
	for(int i=1;i<n;i<<=1){
		ll wn=pow_mod(root,type==1?(mod-1)/(i<<1):mod-1-(mod-1)/(i<<1));
		for (int p=i<<1,j=0;j<n;j+=p){
			ll w=1;
			for (int k=0;k<i;k++,(w*=wn)%=mod){
				ll l=x[j+k],r=x[j+k+i]*w%mod;
				x[j+k]=(l+r)%mod;x[j+k+i]=(l-r+mod)%mod;
			}
		}
	}
}
void INV(int n,int L,ll a[N],ll b[N]){
	if (n==1) b[0]=pow_mod(a[0],mod-2);
	else {
		INV(n>>1,L-1,a,b);int len=(n<<1);
		for (int i=0;i<n;i++)c[i]=a[i];
		for (int i=n;i<len;i++)c[i]=0;
		ntt(c,len,L+1,1);ntt(b,len,L+1,1);
		for (int i=0;i<len;i++) b[i]=(2-c[i]*b[i]%mod+mod)%mod*b[i]%mod;
		ntt(b,len,L+1,-1);ll inv=pow_mod(len,mod-2);
		for (int i=0;i<len;i++) b[i]=b[i]*inv%mod;
		for (int i=n;i<len;i++) b[i]=0;
	}
}
int main(){
	for (scanf("%d%d",&t,&m),i=1;i<=m;i++)
		scanf("%d",&x),a[x]--,mx=max(mx,x);
	for (a[0]=1,i=0;i<=mx;i++) a[i]=(a[i]+mod)%mod;
	m=2*mx;for (n=1;n<=m;n<<=1)L++;INV(n,L,a,b);
	printf("%I64d\n",b[t]%mod);
}
```

/*
多项式求逆是基于倍增的

假设我们知道

h(x)f(x)=1(mod x^n)

移项得

(h(x)*f(x)-1)=0(mod x^n)

两边同时求平方得

h(x)^2*f(x)^2 - 2*h(x)*f(x) +1 = 0 (mod x^2n)

设 g(x)*f(x)=1(mod x^2n)

两边同时乘以 g(x)可以得

h(x)^2*f(x) -2*h(x) + g(x) =0 （mod x^2n)

我们移项可以得到

g(x) = h(x) *(2- f(x)*h(x)

插头 dp

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;
unordered_map<ll,ll> f[2];int n,m,s,t,i,j;bool mp[100][100];char ch;
ll pos(int v, int x){return (v << (x << 1));}//返回第 x 大块的 v 状态（两个二进制来状压）
ll work(){
	int now=0,nxt=1;
	f[0][0] = 1;
	ll U=(1LL<<((m + 1)<<1))-1;//全集
	for(int i = 1; i <= n; i++){
		for(int j = 1; j <= m; j++){
			f[nxt].clear();
			for(auto &k:f[now]){
				ll S = k.first, val = k.second;
```

```
                ll L = (S >> ((j - 1) << 1)) & 3, R = (S >> (j << 1)) & 3;//分割线（L 是当前竖着的那个，R 是紧接着横着的那个）
                if(!mp[i][j]) {
                    if(!L && !R) f[nxt][S] += val;
                    continue;
                }// 0 0
                if(!L && !R) {
        /*0 0 -> 1 2*/if(mp[i][j + 1] && mp[i + 1][j]) f[nxt][S ^ pos(1, j - 1) ^ pos(2, j)] += val;
                }//2 1
                else if(L == 2 && R == 1) {//2 1 -> 0 0
                    f[nxt][S ^ pos(L, j - 1) ^ pos(R, j)] += val;
                }// 0 1 // 1 0 // 0 2 // 2 0
                else if(!L || !R) {//0 1 // 0 2
                    if(!L) {//拐弯
                        if(mp[i][j + 1]) f[nxt][S] += val;//不拐弯
                        if(mp[i + 1][j]) f[nxt][S ^ pos(L, j - 1) ^ pos(L, j) ^ pos(R, j - 1) ^ pos(R, j)] += val;
                    }//同上
                    else if(!R) {
                        if(mp[i][j + 1]) f[nxt][S ^ pos(L, j - 1) ^ pos(L, j) ^ pos(R, j - 1) ^ pos(R, j)] += val;

                        if(mp[i + 1][j]) f[nxt][S] += val;
                    }
                }//1 1 // 2 2
                else if(L == R) {// 1 1
                    if(L == 1) {
                        int du = 0;//1 1 -> 0 0 但是源头接口处右插头变成左插头
                        for(int p = j; ; p++) {
                            ll o = (S >> (p << 1)) & 3;
                            if(o == 1) du++;
                            if(o == 2) du--;
                            if(!du) {//原来状态消去，弄上新状态
                                f[nxt][S ^ pos(L, j - 1) ^ pos(R, j) ^ pos(2, p) ^ pos(1, p)] += val;

                                break;
                            }
                        }
                    }//根上面差不多
                    else if(L == 2) {
                        int du = 0;
                        for(int p = j - 1; ; p--) {
                            ll o = (S >> (p << 1)) & 3;
                            if(o == 1) du--;
                            if(o == 2) du++;
```

```
                                    if(!du) {
                                        f[nxt][S ^ pos(L, j - 1) ^ pos(R, j) ^ pos(2, p) ^ pos(1, p)] +=
val;
                                        break;
                                    }
                                }
                            }
                        }//本状态当且仅当是终点，用于封口
                        else if(L == 1 && R == 2 && i == s && j == t) return val;
                    }
                    swap(now, nxt);
                }
                f[nxt].clear();//末尾的竖分割线到了下一行就变成了行首的分割线，把状态《《给分
割线腾出地方
                for(auto &k:f[now]) f[nxt][(k.first << 2) & U] += k.second;
                swap(now, nxt);
            }
            return 0;
}
int main() {
            scanf("%d%d",&n,&m);
            for(i = 1;i<= n;i++)
                for(j = 1; j <= m; j++){
                    scanf(" %c",&ch);
                    mp[i][j]=ch=='.';
                    if(mp[i][j]) s = i, t = j;
                }
            printf("%lld\n", work());
            return 0;
}
/*
4 4
**..
....
....
....

2
*/ )
```

支配树

```
#include<bits/stdc++.h>
using namespace std;const int N=3e5+7;
struct data{int to,next;}e[N<<3];int cnt;
struct graph{
	int head[N];
	void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
}g,rg,c,g1;int n,m,i,j,x,y,res,fa[N],pre[N],size[N],semi[N],idom[N],mn[N],dfn[N],ind,id[N];
int find(int x){
	if(x==fa[x])return x;
	int t=fa[x];fa[x]=find(fa[x]);
	if(dfn[semi[mn[t]]]<dfn[semi[mn[x]]])mn[x]=mn[t];
	return fa[x];
}
void dfs(int x){
	id[dfn[x]=++ind]=x;
	for(int i=g.head[x];i;i=e[i].next)if(!dfn[e[i].to])
		pre[e[i].to]=x,dfs(e[i].to);
}
void tarjan(){
	for(dfs(1),i=ind;i>=2;--i){
		for(res=n,x=id[i],j=rg.head[x];j;j=e[j].next)if(dfn[e[j].to]){
			find(e[j].to);
			res=min(res,dfn[semi[mn[e[j].to]]]);
		}
		fa[x]=pre[x];semi[x]=id[res];c.ins(semi[x],x);y=id[i-1];
		for(j=c.head[y];j;j=e[j].next){
			x=e[j].to;find(x);
			if(semi[mn[x]]==y)idom[x]=y;
			else idom[x]=mn[x];
		}
	}
	for(i=2;i<=ind;++i){
		x=id[i];
		if(idom[x]!=semi[x])idom[x]=idom[idom[x]];
	}
}
void dfs1(int x)
{size[x]=1;for(int i=g1.head[x];i;i=e[i].next)dfs1(e[i].to),size[x]+=size[e[i].to];}
int main(){
	for(scanf("%d%d",&n,&m);m--;)scanf("%d%d",&i,&j),g.ins(i,j),rg.ins(j,i);
	for(i=1;i<=n;++i)fa[i]=semi[i]=mn[i]=i;
```

```
	tarjan();for(i=2;i<=n;++i)g1.ins(idom[i],i);
	for(dfs1(1),i=1;i<=n;++i)printf("%d ",size[i]);
}
```

支配树转灭绝树

```
#include<bits/stdc++.h>
using namespace std;const int N=3e5+7;
struct data{int to,next;}e[N*20];int cnt;
struct graph{
	int head[N];
	void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
}g,rg,g1,g2,g3;int
n,m,i,j,k,res,x,y,t,w,q[N],size[N],dfn[N],in[N],h[N],id[N],ind,f[N][20],pre[N],fa[N],mn[N],semi[N];
int find(int x){
	if(x==fa[x])return x;
	int t=fa[x];fa[x]=find(fa[x]);
	if(dfn[semi[mn[t]]]<dfn[semi[mn[x]]])mn[x]=mn[t];
	return fa[x];
}
void dfs(int x){
	id[dfn[x]=++ind]=x;if(pre[x])g1.ins(pre[x],x),in[x]++,g2.ins(x,pre[x]);
	for(int i=g.head[x];i;i=e[i].next)if(!dfn[e[i].to])
		pre[e[i].to]=x,dfs(e[i].to);
}
int lca(int x,int y){
	if(h[x]<h[y])swap(x,y);int t=h[x]-h[y];
	for(int i=19;i>=0;--i)if(t>>i&1)x=f[x][i];
	for(int i=19;i>=0;--i)if(f[x][i]!=f[y][i])x=f[x][i],y=f[y][i];
	return x==y?x:f[x][0];
}
void dfs1(int x){size[x]=1;for(int i=g3.head[x];i;i=e[i].next)dfs1(e[i].to),size[x]+=size[e[i].to];}
int main(){
	for(scanf("%d%d",&n,&m),i=1;i<=m;++i)scanf("%d%d",&x,&y),g.ins(x,y),rg.ins(y,x);
	for(i=1;i<=n;++i)fa[i]=mn[i]=semi[i]=i;
	for(dfs(1),i=ind;i>=2;--i){
		for(res=n,x=id[i],j=rg.head[x];j;j=e[j].next)if(dfn[e[j].to]){
			if(dfn[e[j].to]<dfn[x])res=min(res,dfn[e[j].to]);
			else find(e[j].to),res=min(res,dfn[semi[mn[e[j].to]]]);
```

```
            }
            fa[x]=pre[x];semi[x]=id[res];g1.ins(id[res],x);in[x]++;g2.ins(x,id[res]);
        }
        for(i=1;i<=n;++i)if(!in[i]&&dfn[i])q[w++]=i;
        for(;t!=w;){
            x=q[t++];for(i=g1.head[x];i;i=e[i].next)if(--in[e[i].to]==0)q[w++]=e[i].to;
        }
        for(i=0;i<w;++i){
            x=q[i];j=g2.head[x];k=e[j].to;
            for(j=e[j].next;j;j=e[j].next)k=lca(k,e[j].to);
            h[x]=h[k]+1;
            f[x][0]=k;
            g3.ins(k,x);
            for(j=1;j<=19;++j)f[x][j]=f[f[x][j-1]][j-1];
        }for(dfs1(1),i=1;i<=n;++i)printf("%d ",size[i]);
}
```

李超线段树斜率优化

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
#define fr first
#define sc second
using namespace std;const int N=3e5+7;typedef long long ll;typedef pair<int,int>pa;
struct node{
    ll K,B;
    ll operator[](ll x){return x*K+B;}
}t[N<<2];int n,i,j,ATK,d[N],tot;ll pre[N],suf[N],ans,res=1e18;pa s[N];
bool cmp(pa a,pa b){return a.fr*b.sc>a.sc*b.fr;}
int find(int x){return lower_bound(d+1,d+tot+1,x)-d;}
bool hei(node a,node b,ll x){return a[x]>b[x];}
void update(int rt,int l,int r,node x){
    int
mid=l+r>>1;if(hei(x,t[rt],d[mid]))swap(x,t[rt]);if(l==r||hei(t[rt],x,d[l])&&hei(t[rt],x,d[r]))return;
    if(t[rt].K>x.K)update(lson,l,mid,x);else update(rson,mid+1,r,x);
}
ll query(int rt,int l,int r,int x){
    ll res=t[rt][d[x]];if(l==r)return res;int mid=l+r>>1;
    return max(res,x<=mid?query(lson,l,mid,x):query(rson,mid+1,r,x));
}
node go(int j){return node{-s[j].fr,suf[j+1]*s[j].sc+s[j].fr*(pre[j]-1)};}
```

```
int main(){
	for(scanf("%d%d",&n,&ATK),i=1;i<=n;++i)scanf("%d%d",&s[i].fr,&s[i].sc),d[i]=s[i].sc=ceil((dou
ble)s[i].sc/ATK);
	sort(d+1,d+n+1);tot=unique(d+1,d+n+1)-d-1;
	for(sort(s+1,s+n+1,cmp),i=1;i<=n;++i)pre[i]=pre[i-1]+s[i].sc;
	for(i=n;i>=1;--i)suf[i]=s[i].fr+suf[i+1];
	for(i=1;i<=n;++i)ans+=(pre[i]-1)*s[i].fr;
	for(update(1,1,tot,go(n)),i=n-1;i>=1;--i)
		res=min(res,ans-((pre[i]-1)*s[i].fr+s[i].sc*suf[i+1]+query(1,1,tot,find(s[i].sc)))),
		update(1,1,tot,go(i));
	printf("%lld\n",res);
}
```

虚树

```
#include<bits/stdc++.h>
using namespace std;const int N=25*(1e4+7);typedef long long ll;
struct data{int to,next,v;}e[N*4];int fa[N][20],d[N],q[N],a[N],cnt,pos[N],ind,n,m,K,i,j,top,tot,x,y,z;ll
f[N],mn[N];
struct graph{
	int head[N];
	void ins(int u,int v,int
w=0){if(u==v)return;e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
}g1,g2;
bool cmp(int x,int y){return pos[x]<pos[y];}
void dfs(int x){
	pos[x]=++ind;
	for(int i=1;(1<<i)<=d[x];++i)fa[x][i]=fa[fa[x][i-1]][i-1];
	for(int i=g1.head[x];i;i=e[i].next)if(e[i].to!=fa[x][0])
		mn[e[i].to]=min(mn[x],(ll)e[i].v),fa[e[i].to][0]=x,
		d[e[i].to]=d[x]+1,dfs(e[i].to);
}
int lca(int x,int y){
	if(d[x]<d[y])swap(x,y);
	int t=d[x]-d[y];
	for(int i=19;i>=0;--i)if(t>>i&1)x=fa[x][i];
	for(int i=19;i>=0;--i)if(fa[x][i]!=fa[y][i])x=fa[x][i],y=fa[y][i];
	return x==y?x:fa[x][0];
}
void dp(int x){
	f[x]=mn[x];ll tmp=0;
	for(int i=g2.head[x];i;i=e[i].next)dp(e[i].to),tmp+=f[e[i].to];
```

```
	g2.head[x]=0;
	if(tmp==0)f[x]=mn[x];else f[x]=min(f[x],tmp);
}
int main(){
	for(scanf("%d",&n),i=1;i<n;++i)scanf("%d%d%d",&x,&y,&z),g1.ins(x,y,z),g1.ins(y,x,z);
	for(mn[1]=1e18,dfs(1),scanf("%d",&m);m--;cnt=0){
		for(scanf("%d",&K),i=1;i<=K;++i)scanf("%d",a+i);sort(a+1,a+K+1,cmp);
		for(tot=1,i=2;i<=K;++i)if(lca(a[i],a[tot])!=a[tot])a[++tot]=a[i];
		for(q[top=1]=1,i=1;i<=tot;++i){
			for(j=lca(a[i],q[top]);;){
				if(d[j]>=d[q[top-1]]){
					g2.ins(j,q[top--]);
					if(q[top]!=j)q[++top]=j;
					break;
				}
				g2.ins(q[top-1],q[top]);top--;
			}if(q[top]!=a[i])q[++top]=a[i];
		}while(--top)g2.ins(q[top],q[top+1]);dp(1);printf("%lld\n",f[1]);
	}
}
```

长链剖分

```
#include<bits/stdc++.h>
using namespace std;
const int N=100010;typedef long long ll;
struct data{int to,next;}e[N<<1];int head[N],cnt,n,d[N],son[N],fa[N],i,a,b;ll
ans,mem[N*10],*f[N],*g[N],*now=mem+1;
void ins(int u,int v){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;}
void dfs1(int x){
	d[x]=0;
	for(int i=head[x];i;i=e[i].next) if(e[i].to!=fa[x]){
		fa[e[i].to]=x,dfs1(e[i].to),d[x]=max(d[x],d[e[i].to]+1);
		if(d[e[i].to]>d[son[x]])son[x]=e[i].to;
	}
}
void dfs2(int x){
	int i,j,y;
	if(son[x])f[son[x]]=f[x]+1,g[son[x]]=g[x]-1,dfs2(son[x]);
	f[x][0]=1,ans+=g[x][0];
	for(i=head[x];i;i=e[i].next)if(e[i].to!=fa[x]&&e[i].to!=son[x]){
```

```
        y=e[i].to,f[y]=now,now+=d[y]+1,g[y]=now+d[y]+1,now+=d[y]*2+2,dfs2(y);
        for(j=d[y];j>=0;j--){
            if(j)   ans+=f[x][j-1]*g[y][j];
            ans+=g[x][j+1]*f[y][j];
            g[x][j+1]+=f[x][j+1]*f[y][j];
        }
        for(j=0;j<=d[y];j++){
            if(j)g[x][j-1]+=g[y][j];
            f[x][j+1]+=f[y][j];
        }
    }
}
int main(){
    scanf("%d",&n);d[0]=-1;
    for(i=1;i<n;i++)scanf("%d%d",&a,&b),ins(a,b),ins(b,a);dfs1(1);
    f[1]=now,now+=d[1]+1,g[1]=now+d[1]+1,now+=d[1]*2+2,dfs2(1);
    printf("%lld",ans);
    return 0;
}
```

Manacher 算法

```
#include<bits/stdc++.h>
using namespace std;const int N=2e6+7;
char s[N],t[N];int mx,p,T_T,tot,len[N],i,n;
int manacher(int res=0){
    mx=p=tot=0;t[0]='-';t[++tot]='*';for(i=1;i<=n;++i)t[++tot]=s[i],t[++tot]='*';t[tot+1]='+';
    for(i=1;i<=tot;++i){
        if(mx>i)len[i]=min(mx-i,len[2*p-i]);else len[i]=1;
        while(t[i-len[i]]==t[i+len[i]])len[i]++;
        if(len[i]+i>mx)mx=len[i]+i,p=i;
        res=max(res,len[i]-1);
    }return res;
}
int main(){
    for(scanf("%d",&T_T);T_T--;)scanf("%s",s+1),n=strlen(s+1),printf("%d\n",manacher());
}
```

线段树查询区间历史最值

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
using namespace std;const int N=1e5+7,inf=-0x3f3f3f3f;
struct data{
	int x,y;
	data(int x=0,int y=inf):x(x),y(y){}
	friend data operator+(data a,data b){return data(max(inf,a.x+b.x),max(b.y,a.y+b.x));}
	friend data operator*(data a,data b){return data(max(a.x,b.x),max(a.y,b.y));}
}pt[N<<2],nt[N<<2];int pmx[N<<2],nmx[N<<2],a[N],n,m,l,r,x,i,j;char ch;
void add(int rt,data n,data p){
	pt[rt]=pt[rt]*(nt[rt] + p);
	nt[rt]=nt[rt]+n;
	pmx[rt]=max(pmx[rt],max(nmx[rt]+p.x,p.y));
	nmx[rt]=max(nmx[rt]+n.x,n.y);
}
void update(int rt){pmx[rt]=max(pmx[lson],pmx[rson]);nmx[rt]=max(nmx[lson],nmx[rson]);}
void pushdown(int rt){
	add(lson,nt[rt],pt[rt]);add(rson,nt[rt],pt[rt]);
	nt[rt]=pt[rt]=data();
}
void build(int rt,int l,int r){
	if(l==r){pmx[rt]=nmx[rt]=a[l];return;}int mid=l+r>>1;
	build(lson,l,mid);build(rson,mid+1,r);update(rt);
}
void modify(int rt,int l,int r,int a,int b,data A){
	if(a<=l&&r<=b){add(rt,A,A);return;}int mid=l+r>>1;pushdown(rt);
	if(a<=mid)modify(lson,l,mid,a,b,A);if(b>mid)modify(rson,mid+1,r,a,b,A);update(rt);
}
int query(int rt,int l,int r,int a,int b,int val){
	if(a<=l&&r<=b)return val==0?nmx[rt]:pmx[rt];pushdown(rt);int mid=l+r>>1,res=inf;
	if(a<=mid)res=query(lson,l,mid,a,b,val);if(b>mid)res=max(res,query(rson,mid+1,r,a,b,val));re
turn res;
}
int main(){
	for(scanf("%d",&n),i=1;i<=n;++i)scanf("%d",a+i);build(1,1,n);
	for(scanf("%d",&m);m--;){
		scanf(" %c%d%d",&ch,&l,&r);
		if(ch=='Q')printf("%d\n",query(1,1,n,l,r,0));
		if(ch=='A')printf("%d\n",query(1,1,n,l,r,1));
```

```
            if(ch=='P')scanf("%d",&x),modify(1,1,n,l,r,data(x,inf));
            if(ch=='C')scanf("%d",&x),modify(1,1,n,l,r,data(inf,x));
        }
}
```

线段树区间最小值覆盖

```
#include<bits/stdc++.h>
#define lson (rt<<1)
#define rson (rt<<1|1)
using namespace std;const int N=1e6+7;typedef long long ll;
ll sum[N<<2],mx[N<<2],mx1[N<<2],z,cnt[N<<2],tag[N<<2];int T_T,n,m,op,l,r,i,j,a[N];
void add(int rt,ll val){
    if(mx[rt]<val)return;
    sum[rt]-=cnt[rt]*(mx[rt]-val);
    tag[rt]=mx[rt]=val;
}
void update(int rt){
    if(mx[lson]==mx[rson]){
        mx[rt]=mx[lson];cnt[rt]=cnt[lson]+cnt[rson];
        mx1[rt]=max(mx1[lson],mx1[rson]);
    }else if(mx[lson]<mx[rson]){
        mx[rt]=mx[rson];cnt[rt]=cnt[rson];
        mx1[rt]=max(mx1[rson],mx[lson]);
    }else {
        mx[rt]=mx[lson];cnt[rt]=cnt[lson];
        mx1[rt]=max(mx1[lson],mx[rson]);
    }
    sum[rt]=sum[lson]+sum[rson];
}
void pushdown(int rt){if(tag[rt]!=-1)add(lson,tag[rt]),add(rson,tag[rt]),tag[rt]=-1;}
void build(int rt,int l,int r){
    tag[rt]=-1;if(l==r){sum[rt]=mx[rt]=a[l];cnt[rt]=1;mx1[rt]=-1;return;}int mid=l+r>>1;
    build(lson,l,mid);build(rson,mid+1,r);update(rt);
}
void modify(int rt,int l,int r,int a,int b,ll val){
    if(mx[rt]<=val)return;
    if(a<=l&&r<=b&&mx1[rt]<val){add(rt,val);return;}pushdown(rt);int mid=l+r>>1;
    if(a<=mid)modify(lson,l,mid,a,b,val);if(b>mid)modify(rson,mid+1,r,a,b,val);update(rt);
}
ll query1(int rt,int l,int r,int a,int b){
```

```
	if(a<=l&&r<=b)return sum[rt];int mid=l+r>>1;ll res=0;pushdown(rt);
	if(a<=mid)res=query1(lson,l,mid,a,b);if(b>mid)res+=query1(rson,mid+1,r,a,b);return res;
}
ll query2(int rt,int l,int r,int a,int b){
	if(a<=l&&r<=b)return mx[rt];int mid=l+r>>1;ll res=-1;pushdown(rt);
	if(a<=mid)res=query2(lson,l,mid,a,b);if(b>mid)res=max(query2(rson,mid+1,r,a,b),res);return
res;
}
int main(){
	for(scanf("%d",&T_T);T_T--;){
		for(scanf("%d%d",&n,&m),i=1;i<=n;++i)scanf("%d",a+i);build(1,1,n);
		for(;m--;){
			scanf("%d%d%d",&op,&l,&r);
			if(op==0)scanf("%lld",&z),modify(1,1,n,l,r,z);
			else if(op==1)printf("%lld\n",query2(1,1,n,l,r));
			else printf("%lld\n",query1(1,1,n,l,r));
		}
	}
}
```

点分治

```
#include<cstdio>
#include<algorithm>
using namespace std;const int N=1e4+7;
struct data{int to,next,v;}e[N<<1];int
n,K,i,x,y,z,d[N],q[N],cnt,ans,head[N],sum,root,son[N],f[N],vis[N];
void ins(int u,int v,int w){e[++cnt].to=v;e[cnt].next=head[u];head[u]=cnt;e[cnt].v=w;}
void insert(int u,int v,int w){ins(u,v,w);ins(v,u,w);}
void getroot(int x,int fa){
	son[x]=1;f[x]=0;
	for(int i=head[x];i;i=e[i].next)if(!vis[e[i].to]&&e[i].to!=fa){
		getroot(e[i].to,x);
		son[x]+=son[e[i].to];
		f[x]=max(son[e[i].to],f[x]);
	}
	f[x]=max(sum-son[x],f[x]);
	if(f[x]<f[root])root=x;
}
void getdeep(int x,int fa){
	q[++q[0]]=d[x];
	for(int i=head[x];i;i=e[i].next)if(e[i].to!=fa&&!vis[e[i].to]){
```

```
            d[e[i].to]=d[x]+e[i].v;
            getdeep(e[i].to,x);
        }
}
int cal(int x,int now){
        d[x]=now;q[0]=0;getdeep(x,0);
        sort(q+1,q+q[0]+1);
        int L=1,R=q[0],res=0;
        for(;L!=R;){
            if(q[L]+q[R]<=K)res+=(R-L),L++;
            else R--;
        }
        return res;
}
void work(int x){
        vis[x]=1;ans+=cal(x,0);
        for(int i=head[x];i;i=e[i].next)if(!vis[e[i].to])ans-=cal(e[i].to,e[i].v);
        for(int i=head[x];i;i=e[i].next)if(!vis[e[i].to]){
            sum=son[e[i].to];root=0;getroot(e[i].to,0);work(root);
        }
}
int main(){
        for(f[0]=N;scanf("%d%d",&n,&K),n+K;){
            for(i=1;i<n;++i)scanf("%d%d%d",&x,&y,&z),insert(x,y,z);
            sum=n;root=0;getroot(1,0);work(root);printf("%d\n",ans);
            for(i=cnt=ans=0;i<=n;++i)head[i]=vis[i]=0;
        }
}
```

求 1 到 n 中素数的个数 min25 筛

```
#include<bits/stdc++.h>
#define int long long
using namespace std;const int N=1e6+7;
int q[N],tot,i,j,w[N],g[N],Sqr,id1[N],id2[N],n,v[N],m;
void solve(){
        Sqr=sqrt(n);memset(v,0,sizeof(v));tot=m=0;
        for(i=2;i<=Sqr;++i){
            if(!v[i])q[++tot]=i;
            for(j=1;j<=tot&&i*q[j]<=Sqr;++j){
```

```
                v[i*q[j]]=1;if(i%q[j]==0)break;
            }
        }
        for(i=1;i<=n;i=j+1){
            j=n/(n/i);w[++m]=n/i;
            if(w[m]<=Sqr)id1[w[m]]=m;else id2[n/w[m]]=m;
            g[m]=w[m]-1;
        }
        for(j=1;j<=tot;++j)
            for(i=1;i<=m&&q[j]*q[j]<=w[i];++i){
                int k=(w[i]/q[j]<=Sqr)?id1[w[i]/q[j]]:id2[n/(w[i]/q[j])];
                g[i]=(g[i]-g[k]+j-1);
            }
        printf("%lld\n",g[1]);
}
int32_t main(){for(;~scanf("%lld",&n);solve());}
```

Min25 筛算法

```
#include<bits/stdc++.h>
using namespace std;typedef long long ll;const int MAX=222222,MOD=1000000007;
ll n,Sqr,w[MAX],pri[MAX],id1[MAX],id2[MAX],h[MAX],g[MAX],m;bool v[MAX];int tot,sp[MAX];
void pre(int n){
        v[1]=true;
        for(int i=2;i<=n;++i){
            if(!v[i])pri[++tot]=i,sp[tot]=(sp[tot-1]+i)%MOD;
            for(int j=1;j<=tot&&i*pri[j]<=n;++j){
                v[i*pri[j]]=true;
                if(i%pri[j]==0)break;
            }
        }
}
int S(ll x,int y){
        if(x<=1||pri[y]>x)return 0;
        int k=(x<=Sqr)?id1[x]:id2[n/x],ret=(g[k]-sp[y-1]-h[k]+y-1)%MOD;
        if(y==1)ret+=2;
        for(int i=y;i<=tot&&1ll*pri[i]*pri[i]<=x;++i){
            ll t1=pri[i],t2=1ll*pri[i]*pri[i];
            for(int e=1;t2<=x;++e,t1=t2,t2*=pri[i])
                (ret+=((1ll*S(x/t1,i+1)*(pri[i]^e)%MOD+(pri[i]^(e+1))%MOD)))%=MOD;
        }
```

```
	return ret;
}
int main(){
	scanf("%lld",&n);Sqr=sqrt(n);pre(Sqr);
	for(ll i=1,j;i<=n;i=j+1){
		j=n/(n/i);w[++m]=n/i;
		h[m]=(w[m]-1)%MOD;
		g[m]=(w[m]%MOD)*((w[m]+1)%MOD)%MOD;
		if(g[m]&1)g[m]=g[m]+MOD;g[m]/=2;g[m]--;
		if(w[m]<=Sqr)id1[w[m]]=m;else id2[j]=m;
	}
	for(int j=1;j<=tot;++j)
		for(int i=1;i<=m&&pri[j]*pri[j]<=w[i];++i){
			int k=(w[i]/pri[j]<=Sqr)?id1[w[i]/pri[j]]:id2[n/(w[i]/pri[j])];
			(g[i]-=1ll*pri[j]*(g[k]-sp[j-1])%MOD)%=MOD;
			(h[i]-=h[k]-j+1)%=MOD;
		}
	int ans=S(n,1)+1;
	printf("%d\n",(ans+MOD)%MOD);
	return 0;
}
```

手动开栈模板

```
#include<bits/stdc++.h>
#include<tr1/unordered_map>
using namespace std;tr1::unordered_map<int,int>mp;
int main(){
	int size=256<<20;
	char*p=(char*)malloc(size)+size;
	cin>>size;
	__asm__("movl %0, %%esp\n" : : "r"(p));
}
```