数据结构

堆

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=1e6+1000;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,opt,a[N],siz=0;
void Insert(int x){
    int now=++siz;
    a[now]=x;
    while(now!=1&&a[now>>1]>a[now]){
        swap(a[now>>1],a[now]);
        now>>=1;
    }
}
void Pop(){
    int now=1,k,nxt;
    k=a[1]=a[siz--];
    while(now<=siz){
        int minn=1<<30;
        if((now<<1)<=siz)minn=a[now<<1],nxt=now<<1;
        if((now<<1|1)<=siz){
            if(a[now<<1|1]<minn){
                minn=a[now<<1|1];
                nxt=now<<1|1;;
            }
        }
        if(minn>=k)return ;
        swap(a[now],a[nxt]);
        now=nxt;
    }
}
```

```
int main()
{
    n=read();
    for(int i=1;i<=n;i++){
        opt=read();
        if(opt==1)Insert(read());
        if(opt==2)printf("%d\n",a[1]);
        if(opt==3)Pop();
    }
    return 0;
}
```

ST表

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=1e6+1000;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,opt,a[N],siz=0;
void Insert(int x){
    int now=++siz;
    a[now]=x;
    while(now!=1&&a[now>>1]>a[now]){
        swap(a[now>>1],a[now]);
        now>>=1;
    }
}
void Pop(){
    int now=1,k,nxt;
    k=a[1]=a[siz--];
    while(now<=siz){
        int minn=1<<30;
        if((now<<1)<=siz)minn=a[now<<1],nxt=now<<1;
        if((now<<1|1)<=siz){
            if(a[now<<1|1]<minn){
                minn=a[now<<1|1];
                nxt=now<<1|1;;
            }
        }
        if(minn>=k)return ;
        swap(a[now],a[nxt]);
        now=nxt;
    }
}
int main()
{
    n=read();
    for(int i=1;i<=n;i++){
        opt=read();
        if(opt==1)Insert(read());
        if(opt==2)printf("%d\n",a[1]);
        if(opt==3)Pop();
    }
    return 0;
}
```

并查集

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=10009;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,opt,x,y,fa[N];
int fid(int x){return (fa[x]==x)?x:(fa[x]=fid(fa[x]));}
int main()
{
    n=read();m=read();
    for(int i=1;i<=n;i++)fa[i]=i;
    for(int i=1;i<=m;i++){
        opt=read();x=read();y=read();
        if(opt==1)fa[fid(x)]=fid(y);
        else printf("%s\n",(fid(x)==fid(y))?"Y":"N");
    }
    return 0;
}
```

树状数组

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=6e5+1000;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,a[N];
void add(int x,int k){for(;x<=n;x+=x&-x)a[x]+=k;}
int ask(int x){
    int ans=0;
    for(;x;x-=x&-x)ans+=a[x];
    return ans;
}
int main()
{
    n=read();m=read();
    for(int i=1;i<=n;i++)add(i,read());
    for(int i=1;i<=m;i++){
        int opt=read(),x=read();
        if(opt==1)add(x,read());
        else printf("%d\n",ask(read())-ask(x-1));
    }
    return 0;
}
```

线段树

```
#include <bits/stdc++.h>
using namespace std;
long long read(){
    char c;long long num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
long long n,m,tree[100000*5],a[100009],add[100000*5],opt,x,y;
void build(int l,int r,int rt){
    if(l==r){tree[rt]=a[l];return ;}
    build(l,(l+r)>>1,rt<<1);
    build(((l+r)>>1)+1,r,rt<<1|1);
    tree[rt]=tree[rt<<1]+tree[rt<<1|1];
}
void pushdown(int l,int r,int rt){
    int mid=(l+r)>>1;
    add[rt<<1]+=add[rt];
    tree[rt<<1]+=(mid-l+1)*add[rt];
    add[rt<<1|1]+=add[rt];
    tree[rt<<1|1]+=(r-mid)*add[rt];
    add[rt]=0;
}
void change(int l,int r,int L,int R,long long k,int rt){
    if(L<=l&&r<=R){add[rt]+=k;tree[rt]+=(r-l+1)*k;return;}
    int mid=(l+r)>>1;
    pushdown(l,r,rt);
    if(L<=mid)change(l,mid,L,R,k,rt<<1);
    if(mid+1<=R)change(mid+1,r,L,R,k,rt<<1|1);
    tree[rt]=tree[rt<<1]+tree[rt<<1|1];
}
long long ask(int l,int r,int L,int R,int rt){
    if(L<=l&&r<=R){return tree[rt];}
    long long mid=(l+r)>>1,ans=0;
    //cout<<l<<"   "<<r<<endl;
    pushdown(l,r,rt);
    if(L<=mid)ans+=ask(l,mid,L,R,rt<<1);
    if(mid+1<=R)ans+=ask(mid+1,r,L,R,rt<<1|1);
    return ans;
}
int main()
{
    n=read();m=read();
    for(int i=1;i<=n;i++)a[i]=read();
    build(1,n,1);
    for(int i=1;i<=m;i++){
        opt=read();x=read();y=read();
        if(opt==1)change(1,n,x,y,read(),1);
        if(opt==2)printf("%lld\n",ask(1,n,x,y,1));
    }
    return 0;
}
```

数论算法

拓展欧几里得算法

用于求解ax+by=gcd(a,b)不定方程的一组特解。

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int a,b;
int exgcd(int a,int b,int &x,int &y){
    if(b==0){x=1;y=0;return a;}
    int d=exgcd(b,a%b,y,x);
    y-=a/b*x;return d;
}
int main()
{
    int x,y,d;
    a=read();b=read();
    d=exgcd(a,b,x,y);
    cout<<x<<endl;
    //printf("%d\n",((d-b*y)/a%b+b)%b);
    return 0;
}
```

exgcd求逆元

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,p;
void exgcd(int a,int b,int &x,int &y){
    if(b==0){x=1;y=0;return;}
    exgcd(b,a%b,x,y);
    int tmp=x;x=y;y=tmp-a/b*y;
}
int inv(int a){
    int x,y;
    exgcd(a,p,x,y);
    return (x%p+p)%p;
}
int main()
{
    n=read();p=read();
    for(int i=1;i<=n;i++)
        printf("%d\n",inv(i));
    return 0;
}
```

费马小定理求逆元

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,p;
int Pow(int a,int p,int mod){
    int ans=1;
    for(;p;p>>=1,a=1ll*a*a%mod)
        if(p&1)ans=1ll*ans*a%mod;
    return ans;
}
int inv(int x){
    return Pow(x,p-2,p);
}
int main()
```

```
{
    n=read();p=read();
    for(int i=1;i<=n;i++)
        printf("%d\n",inv(i));
    return 0;
}
```

线性求逆元

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,p;
int Pow(int a,int p,int mod){
    int ans=1;
    for(;p;p>>=1,a=1ll*a*a%mod)
        if(p&1)ans=1ll*ans*a%mod;
    return ans;
}
int inv(int x){
    return Pow(x,p-2,p);
}
int main()
{
    n=read();p=read();
    for(int i=1;i<=n;i++)
        printf("%d\n",inv(i));
    return 0;
}
```

Miller_Rabin算法

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
```

```
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int test[109]={2,3,5,7,11,61,24251,13,17,23};
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n;
int Pow(int a,int p,int mod){
    int ans=1;
    for(;p;p>>=1,a=1ll*a*a%mod)
        if(p&1)ans=1ll*ans*a%mod;
    return ans;
}
bool check(int P){
    if(P==1)return 0;
    int k=0,t=P-1;
    while(!(k&1))k++,t>>=1;
    for(int i=0;i<10;i++){
        if(P==test[i])return 1;
        int a=Pow(test[i],t,P),nxt;
        for(int j=1;j<=k;j++){
            nxt=(1ll*a*a)%P;
            if(nxt==1&&a!=1&&a!=P-1)return 0;
            a=nxt;
        }
        if(a!=1)return 0;
    }
    return 1;
}
int main()
{
    read();n=read();
    for(int i=1;i<=n;i++)printf("%s\n",check(read())?"Yes":"No");
    return 0;
}
```

埃氏筛法

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
```

```
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=1e7+1000;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m;
bool f[N];
void init(){
    for(int i=2;i<=n;i++){
        if(f[i])continue;
        for(int j=i+i;j<=n;j+=i)
            f[j]=1;
    }
}
int main()
{
    n=read();m=read();
    f[1]=1;init();
    for(int i=1;i<=m;i++)
        printf("%s\n",f[read()]?"No":"Yes");
    return 0;
}
```

欧拉筛

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=1e7+1000;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,phi[N],cnt=0;
```

```
bool f[N];
void init(){
    for(int i=2;i<=n;i++){
        if(!f[i])phi[++cnt]=i;
        for(int j=1;j<=cnt&&i*phi[j]<=n;j++){
            f[i*phi[j]]=1;
            if(!(i%phi[j]))break;
        }
    }
}
int main()
{
    n=read();m=read();
    f[1]=1;init();
    for(int i=1;i<=m;i++)
        printf("%s\n",f[read()]?"No":"Yes");
    return 0;
}
```

Dijkstra

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <queue>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=100009,M=200009;
struct Node{
    int id,val;
    Node(int a=0,int b=0){id=a;val=b;}
};
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
bool operator <(const Node a,const Node b){return a.val>b.val;}
int n,m,s,dis[N];
int head[N],edge[M],nxt[M],ver[M],tot=1;
priority_queue<Node>q;
void add(int u,int v,int w){
    ver[++tot]=v;edge[tot]=w;nxt[tot]=head[u];head[u]=tot;
}
void dijkstra(){
    memset(dis,0x3f,sizeof(dis));
    dis[s]=0;q.push(Node(s,0));
    while(q.size()){
        Node a=q.top();q.pop();
        int w=a.val,x=a.id;
        if(w>dis[x])continue;
        for(int i=head[x];i;i=nxt[i]){
            int y=ver[i];
            if(dis[y]>dis[x]+edge[i]){
                dis[y]=dis[x]+edge[i];
                q.push(Node(y,dis[y]));
            }
        }
    }
}
int main()
{
    n=read();m=read();s=read();
    for(int i=1;i<=m;i++){
        int u=read(),v=read(),w=read();
        add(u,v,w);
    }
    dijkstra();
    for(int i=1;i<=n;i++)
        printf("%d ",dis[i]);
    printf("\n");
    return 0;
}
```

SPFA

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int inf=2147483647;
const int N=10009,M=500009*2;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,s,d[N],vis[N],q[N*3],hh,tt,lim=N*3-10;
int head[N],nxt[M],edge[M],ver[M],tot=1;
void add(int u,int v,int w){
    ver[++tot]=v;nxt[tot]=head[u];head[u]=tot;edge[tot]=w;
}
void spfa(){
    for(int i=1;i<=n;i++)d[i]=(i==s)?0:inf;
    hh=1;tt=0;q[++tt]=s;
    while(tt!=hh-1){
        int x=q[hh];vis[x]=0;
        hh=(hh==lim-1)?0:hh+1;
        for(int i=head[x];i;i=nxt[i]){
            int y=ver[i];
            if(d[y]>d[x]+edge[i]){
                d[y]=d[x]+edge[i];
                if(!vis[y]){
                    tt=(tt==lim-1)?0:tt+1;
                    q[tt]=y;
                    vis[y]=1;
                }
            }
        }
    }
}
int main()
{
    n=read();m=read();s=read();
    for(int i=1;i<=m;i++){
        int u=read(),v=read(),w=read();
        add(u,v,w);
    }
    spfa();
    for(int i=1;i<=n;i++)
        printf("%d ",d[i]);
    printf("\n");
    return 0;
}
```

LCA

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=500009,M=500009*2;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,fa[23][N],de[N],s;
int head[N],ver[M],nxt[M],tot=1;
void add(int u,int v){
    ver[++tot]=v;nxt[tot]=head[u];head[u]=tot;
    ver[++tot]=u;nxt[tot]=head[v];head[v]=tot;
}
void dfs(int x,int pre){
    for(int i=1;i<23;i++)
        fa[i][x]=fa[i-1][fa[i-1][x]];
    for(int i=head[x];i;i=nxt[i]){
        if(ver[i]==pre)continue;
        de[ver[i]]=de[x]+1;
        fa[0][ver[i]]=x;
        dfs(ver[i],x);
    }
}
void Swap(int &x,int &y){x^=y;y^=x;x^=y;}
int lca(int x,int y){
    if(de[x]<de[y])Swap(x,y);
    int dis=de[x]-de[y];
    for(int i=0;i<23;i++)
        if((1<<i)&dis)x=fa[i][x];
    if(x==y)return x;
    for(int i=22;i>=0;i--)
        if(fa[i][x]!=fa[i][y])
            x=fa[i][x],y=fa[i][y];
    return fa[0][x];
}
int main()
{
    n=read();m=read();s=read();
    for(int i=1;i<n;i++)add(read(),read());
    de[s]=1;dfs(s,s);
    for(int i=1;i<=m;i++)
        printf("%d\n",lca(read(),read()));
    return 0;
}
```

二分图匹配

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
```

```cpp
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int inf=(1<<31)-1;
const int N=1000*10,M=6000009;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,e,s,t,d[N],q[N*4],tt,hh;
int head[N],ver[M],nxt[M],edge[M],tot=1;
void add(int u,int v,int w){
    ver[++tot]=v;nxt[tot]=head[u];head[u]=tot;edge[tot]=w;
    ver[++tot]=u;nxt[tot]=head[v];head[v]=tot;edge[tot]=0;
}
bool bfs(){
    memset(d,0,sizeof(d));
    hh=1;tt=0;q[++tt]=s;d[s]=1;
    while(hh<=tt){
        int x=q[hh];hh++;
        for(int i=head[x];i;i=nxt[i]){
            int y=ver[i];
            if(d[y]||!edge[i])continue;
            d[y]=d[x]+1;
            if(y==t)return 1;
            q[++tt]=y;
        }
    }
    return 0;
}
int dinic(int x,int flow){
    if(x==t)return flow;
    int res=flow,k=0;
    for(int i=head[x];i&&res;i=nxt[i]){
        if(d[ver[i]]!=d[x]+1||!edge[i])continue;
        int y=ver[i];
        k=dinic(y,min(edge[i],res));
        if(!k)d[y]=0;
        edge[i]-=k;
        edge[i^1]+=k;
        res-=k;
    }
    return flow-res;
}
int main()
{
    n=read();m=read();e=read();
    s=n+m+2;t=n+m+4;
    for(int i=1;i<=e;i++){
        int u=read(),v=read();
        if(v>m)continue;
        add(u,v+n,1);
    }
    for(int i=1;i<=n;i++)add(s,i,1);
    for(int i=1;i<=m;i++)add(n+i,t,1);
    int flow,maxflow=0;
    while(bfs())while(flow=dinic(s,inf))maxflow+=flow;
    printf("%d\n",maxflow);
    return 0;
}
```

最大流

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=10009*3,M=100009*3,inf=(1<<31)-1;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,s,t,d[N],q[N*10],tt,hh;
int head[N],nxt[M],ver[M],edge[M],tot=1;
void add(int u,int v,int w){
    ver[++tot]=v;nxt[tot]=head[u];head[u]=tot;edge[tot]=w;
    ver[++tot]=u;nxt[tot]=head[v];head[v]=tot;edge[tot]=0;
}
bool bfs(){
    memset(d,0,sizeof(d));
    hh=1;tt=0;d[s]=1;q[++tt]=s;
    while(hh<=tt){
        int x=q[hh++];
        for(int i=head[x];i;i=nxt[i]){
            int y=ver[i];
            if(d[y]||!edge[i])continue;
            d[y]=d[x]+1;
            if(y==t)return 1;
            q[++tt]=y;
        }
    }
    return 0;
}
int dinic(int x,int flow){
    if(x==t)return flow;
    int res=flow,k;
    for(int i=head[x];i&&res;i=nxt[i]){
        if(d[ver[i]]!=d[x]+1||!edge[i])continue;
        int y=ver[i];
        k=dinic(y,min(edge[i],res));
        if(!k)d[y]=0;
        edge[i]-=k;
        edge[i^1]+=k;
        res-=k;
    }
    return flow-res;
}
int main()
{
    n=read();m=read();
    s=read();t=read();
    for(int i=1;i<=m;i++){
        int u=read(),v=read(),w=read();
        add(u,v,w);
    }
    int flow,maxflow=0;
    while(bfs())while(flow=dinic(s,inf))maxflow+=flow;
    printf("%d\n",maxflow);
    return 0;
}
```

割点

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=20009,M=100009;
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
```

```
int n,m,cnt=0,low[N],cut[N];
int head[N],nxt[M*3],ver[M*3],tot=1,dfn[N],vis[N];
void add(int u,int v){
    ver[++tot]=v;nxt[tot]=head[u];head[u]=tot;
    ver[++tot]=u;nxt[tot]=head[v];head[v]=tot;
}
void tarjan(int x,int fa){
    low[x]=dfn[x]=++cnt;
    int child=0;
    for(int i=head[x];i;i=nxt[i]){
        int y=ver[i];
        if(!dfn[y]){
            tarjan(y,x);
            low[x]=min(low[x],low[y]);
            if(low[y]>=dfn[x]&&x!=fa)cut[x]=1;
            if(x==fa)child++;
        }else low[x]=min(low[x],dfn[y]);
    }
    if(x==fa&&child>=2)cut[x]=1;
}
int main()
{
    n=read();m=read();
    for(int i=1;i<=m;i++)
        add(read(),read());
    for(int i=1;i<=n;i++)
        if(!dfn[i])tarjan(i,i);
    int num=0;
    for(int i=1;i<=n;i++)
        if(cut[i])num++;
    printf("%d\n",num);
    for(int i=1;i<=n;i++)
        if(cut[i])printf("%d ",i);
    printf("\n");
    return 0;
}
```

缩点

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=100009,M=100009*2;
```

```
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,rn,w[N],low[N],dfn[N],st[N],t=0,sd[N],rw[N];
int head1[N],nxt[M*3],ver[M*3],tot=1,cnt=0,vis[N],f[N],head2[N];
void add1(int u,int v){ver[++tot]=v;nxt[tot]=head1[u];head1[u]=tot;}
void add2(int u,int v){ver[++tot]=v;nxt[tot]=head2[u];head2[u]=tot;}
void tarjan(int x){
    low[x]=dfn[x]=++cnt;
    st[++t]=x;vis[x]=1;
    for(int i=head1[x];i;i=nxt[i]){
        if(!dfn[ver[i]]){
            tarjan(ver[i]);
            low[x]=min(low[x],low[ver[i]]);
        }else if(vis[ver[i]]){
            low[x]=min(low[x],dfn[ver[i]]);
        }
    }
    if(low[x]==dfn[x]){
        int y;
        rn++;
        while(y=st[t--]){
            sd[y]=rn;
            vis[y]=0;
            rw[rn]+=w[y];
            if(y==x)break;
        }
    }
}
void dfs(int x){
    if(f[x])return ;
    int maxn=0;
    for(int i=head2[x];i;i=nxt[i]){
        if(!f[ver[i]])dfs(ver[i]);
        maxn=max(maxn,f[ver[i]]);
    }
    f[x]=rw[x]+maxn;
}
int main()
{
    n=read();m=read();
    for(int i=1;i<=n;i++)w[i]=read();
    for(int i=1;i<=m;i++){
        int u=read(),v=read();
        add1(u,v);
    }
    for(int i=1;i<=n;i++)
        if(!dfn[i])tarjan(i);
    for(int x=1;x<=n;x++)
        for(int i=head1[x];i;i=nxt[i])
            if(sd[ver[i]]!=sd[x])
                add2(sd[ver[i]],sd[x]);
    int maxn=0;
    for(int i=1;i<=rn;i++){
        if(!f[i])dfs(i);
        maxn=max(maxn,f[i]);
    }
    printf("%d\n",maxn);
    return 0;
}
```

最小生成树

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int N=5009,M=200009;
struct edge{
    int u,v,w;
}e[M+100];
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int n,m,fa[N+10],ans=0;
bool cmp(edge a,edge b){return a.w<b.w;}
int fid(int a){return (a==fa[a])?a:(fa[a]=fid(fa[a]));}
int main()
{
    n=read();m=read();
    for(int i=1;i<=n;i++)fa[i]=i;
    for(int i=1;i<=m;i++){
        e[i].u=read();
        e[i].v=read();
        e[i].w=read();
    }
    sort(e+1,e+1+m,cmp);
    for(int i=1;i<=m;i++){
        if(fid(e[i].u)!=fid(e[i].v)){
            ans+=e[i].w;
            fa[fid(e[i].u)]=fid(e[i].v);
        }
    }
    printf("%d\n",ans);
    return 0;
}
```

递推算法

矩阵加速

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int mod=1000000007;
struct Mat{
    int n,m,a[109][109];
    Mat(){
        n=0;m=0;
        memset(a,0,sizeof(a));
    }
    void clear(){memset(a,0,sizeof(a));}
    void print(){
        for(int i=1;i<=n;i++){
            for(int j=1;j<=m;j++)
                printf("%d ",a[i][j]);
            printf("\n");
        }
    }
};
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
Mat operator +(Mat A,Mat B){
    if(A.n!=B.n||A.m!=B.m){cerr<<"Wrong"<<endl;return A;}
    Mat C;C.n=A.n;C.m=B.m;
    for(int i=1;i<=A.n;i++)
        for(int j=1;j<=A.m;j++){
            C.a[i][j]=A.a[i][j]+B.a[i][j];
            if(C.a[i][j]>=mod)C.a[i][j]-=mod;
        }
    return C;
}
Mat operator -(Mat A,Mat B){
    if(A.n!=B.n||A.m!=B.m){cerr<<"Wrong"<<endl;return A;}
    Mat C;C.n=A.n;C.m=B.m;
    for(int i=1;i<=A.n;i++)
        for(int j=1;j<=A.m;j++){
            C.a[i][j]=A.a[i][j]-B.a[i][j];
            if(C.a[i][j]<0)C.a[i][j]+=mod;
        }
    return C;
}
Mat operator *(Mat A,Mat B){
    if(A.m!=B.n){cerr<<"Wrong"<<endl;return A;}
    Mat C;C.n=A.n;C.m=B.m;
    for(int i=1;i<=A.n;i++)
        for(int j=1;j<=B.m;j++)
            for(int k=1;k<=A.m;k++)
                C.a[i][j]=(C.a[i][j]+(1ll*A.a[i][k]*B.a[k][j])%mod)%mod;
    return C;
}

Mat operator ^(Mat A,ll p){
    if(A.n!=A.m){cerr<<"Wrong"<<endl;return A;}
    Mat ans;
    ans.m=ans.n=A.n;
    for(int i=1;i<=ans.n;i++)ans.a[i][i]=1;
    for(;p;p>>=1,A=A*A)
        if(p&1)ans=ans*A;
    return ans;
}
void work(){
    int n=read();
    Mat fib,rec;
    fib.n=1;fib.m=3;
    rec.n=3;rec.m=3;
    fib.a[1][1]=1;fib.a[1][2]=1;fib.a[1][3]=1;
    rec.a[1][1]=0;rec.a[1][2]=0;rec.a[1][3]=1;
    rec.a[2][1]=1;rec.a[2][2]=0;rec.a[2][3]=0;
    rec.a[3][1]=0;rec.a[3][2]=1;rec.a[3][3]=1;
    fib=fib*(rec^(n-1));
    printf("%d\n",fib.a[1][1]);
}
int main()
{
    int Case=read();
    while(Case--)work();
    return 0;
```

```
}
```

矩阵快速幂

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int mod=1000000007;
struct Mat{
    int n,m,a[109][109];
    Mat(){
        n=0;m=0;
        memset(a,0,sizeof(a));
    }
    void clear(){memset(a,0,sizeof(a));}
    void print(){
        for(int i=1;i<=n;i++){
            for(int j=1;j<=m;j++)
                printf("%d ",a[i][j]);
            printf("\n");
        }
    }
};
ll read(){
    char c;ll num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
Mat operator +(Mat A,Mat B){
    if(A.n!=B.n||A.m!=B.m){cerr<<"Wrong"<<endl;return A;}
    Mat C;C.n=A.n;C.m=B.m;
    for(int i=1;i<=A.n;i++)
        for(int j=1;j<=A.m;j++){
            C.a[i][j]=A.a[i][j]+B.a[i][j];
            if(C.a[i][j]>=mod)C.a[i][j]-=mod;
        }
    return C;
}
Mat operator -(Mat A,Mat B){
    if(A.n!=B.n||A.m!=B.m){cerr<<"Wrong"<<endl;return A;}
    Mat C;C.n=A.n;C.m=B.m;
    for(int i=1;i<=A.n;i++)
```

```
        for(int j=1;j<=A.m;j++){
            C.a[i][j]=A.a[i][j]-B.a[i][j];
            if(C.a[i][j]<0)C.a[i][j]+=mod;
        }
    return C;
}
Mat operator *(Mat A,Mat B){
    if(A.m!=B.n){cerr<<"Wrong"<<endl;return A;}
    Mat C;C.n=A.n;C.m=B.m;
    for(int i=1;i<=A.n;i++)
        for(int j=1;j<=B.m;j++)
            for(int k=1;k<=A.m;k++)
                C.a[i][j]=(C.a[i][j]+(1ll*A.a[i][k]*B.a[k][j])%mod)%mod;
    return C;
}

ll k;
Mat operator ^(Mat A,ll p){
    if(A.n!=A.m){cerr<<"Wrong"<<endl;return A;}
    Mat ans;
    ans.m=ans.n=A.n;
    for(int i=1;i<=ans.n;i++)ans.a[i][i]=1;
    for(;p;p>>=1,A=A*A)
        if(p&1)ans=ans*A;
    return ans;
}
int main()
{
    Mat A;
    A.n=A.m=read();
    k=read();
    for(int i=1;i<=A.n;i++)
        for(int j=1;j<=A.n;j++)
            A.a[i][j]=read();
    A=A^k;
    A.print();
    return 0;
}
```

字符串

字符串哈希

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <queue>
```

```
#include <vector>
#define ull unsigned long long
#define ll long long
using namespace std;
const int mod=19260817;
const int base1=131;
const int base2=1331;
int n,ans=0;
char s[1000009];
bool f[mod+5];
int hash1(char *s){
    int len=strlen(s);
    ull k=0;
    for(int i=0;i<len;i++)
        k=k*base1+s[i];
    k%=mod;
    return (int)k;
}
int hash2(char *s){
    int len=strlen(s);
    ull k=0;
    for(int i=0;i<len;i++)
        k=k*base2+s[i];
    k%=mod;
    return (int)k;
}
int read(){
    char c;int num,f=1;
    while(c=getchar(),!isdigit(c))if(c=='-')f=-1;num=c-'0';
    while(c=getchar(), isdigit(c))num=num*10+c-'0';
    return f*num;
}
int main()
{
    int k1,k2;
    n=read();
    for(int i=1;i<=n;i++){
        scanf("%s",s);
        k1=hash1(s);
        k2=hash2(s);
        if(f[k1]&&f[k2])ans++;
        else f[k1]=f[k2]=1;
    }
    printf("%d\n",n-ans);
    return 0;
}
```

KMP

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

```cpp
using namespace std;
char s1[1000009],s2[1000009];
int kmp[1000100],len1,len2;
void get_next();
void fid();
int main()
{
    cin>>s1+1>>s2+1;
    len1=strlen(s1+1);
    len2=strlen(s2+1);
    get_next();
    fid();
    for(int i=1;i<=len2;i++){
        cout<<kmp[i]<<" ";
    }
    return 0;
}
void get_next(){
    int j=0;
    for(int i=2;i<=len2;i++){
        while(j&&s2[j+1]!=s2[i])j=kmp[j];
        if(s2[j+1]==s2[i])j++;
        kmp[i]=j;
    }
}
void fid(){
    int j=0;
    for(int i=1;i<=len1;i++){
        while(j&&s2[j+1]!=s1[i])j=kmp[j];
        if(s2[j+1]==s1[i])j++;
        if(j==len2){
            cout<<i-len2+1<<endl;
            j=kmp[j];
        }
    }
}
```