

## Table of Contents

<b>Dijkstra</b> .....	1
<b>Code c++</b> .....	1
Dijkstra mảng 2d (theway).....	3
<b>Code pascal</b> .....	5
<b>Kruskal</b> .....	9
<b>Code c++</b> .....	9
<b>Code pascal</b> .....	11
<b>Tazjan</b> .....	14
<b>Code c++</b> .....	14
<b>Code pascal</b> .....	16
<b>Song liên thông</b> .....	19
<b>Code pascal</b> .....	19
<b>Khớp cầu</b> .....	22
<b>Code pascal</b> .....	22
<b>Stack</b> .....	25
<b>Codepascal</b> .....	25
<b>Hash</b> .....	27
<b>Code c++</b> .....	27
<b>Code pascal</b> .....	28
<b>Dequeue</b> .....	29
<b>Euler</b> .....	31
<b>Floyd</b> .....	32

## Dijkstra

### Code c++

```
#include <cstdio>

#include <iostream>

#include <vector>

#include <queue>

using namespace std;

const int inf=1<<30;

const int gh=100000;

typedef pair<int,int> pii;

struct cmp

{

    bool operator()(pii x, pii y)

    {

        return x.second>y.second;

    }

};

priority_queue<pii, vector<pii>, cmp>heap;

int m,n,s,t;

vector<vector<pii> >dsk;

vector<int> fre,tr,d;

void nhap()

{

    freopen("dijkstra.inp","r",stdin);

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

    cin>>n>>m>>s>>t;

    dsk.resize(gh);fre.resize(gh);d.resize(gh);tr.resize(gh);

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

    {

        d[i]=inf;tr[i]=0;fre[i]=0;

    }

    d[s]=0;
```

```

for (int i=0;i<m;i++)
{
    int u,v,w;
    cin>>u>>v>>w;
    dsk[u].push_back(pii(v,w));
    dsk[v].push_back(pii(u,w));
}
}

void dijkstra()
{
    heap.push(pii(s,0));
    while (!heap.empty())
    {
        int u=heap.top().first;
        heap.pop();
        fre[u]=1;
        for (int i=0;i<dsk[u].size();++i)
        {
            int v=dsk[u][i].first;
            int w=dsk[u][i].second;
            if (fre[v]==0 && d[v]>d[u]+w)
            {
                d[v]=d[u]+w;
                tr[v]=u;
                heap.push(pii(v,d[v]));
            }
        }
    }

    for (int i=1;i<=n;i++) cout<<d[i]<< ' ';
}

int main()
{

```

```

    nhap();
    dijkstra();
    return 0;
}

Dijkstra mảng 2d (theway)
#include <iostream>
#include <cstdio>
#include <vector>
#include <queue>
using namespace std;
const int gh=10000;
const int inf=1<<30;
typedef pair<int,int> pii;
struct diem
{
    int v,w,k;
};

struct cmp
{
    bool operator()(diem x,diem y)
    {
        return x.w>y.w;
    }
};

priority_queue<diem,vector<diem>,cmp>heap;
int m,n,p;
vector<vector<pii> > dsk;
int d[gh][9];
void nhap()
{
    freopen("theway.inp","r",stdin);

```

```

freopen("theway.out","w",stdout);
cin>>n>>m>>p;
dsk.resize(gh);
for (int i=0;i<m;i++)
{
    int u,v,w;
    cin>>u>>v>>w;
    dsk[u].push_back(pii(v,w));
    dsk[v].push_back(pii(u,w));
}
}
void khoitao()
{
    for (int i=1;i<=n;i++)
        for (int j=0;j<=9;j++) d[i][j]=inf;
    d[1][0]=0;
}
void dijkstra()
{
    diem u;
    u.v=1;u.w=0;u.k=0;
    heap.push(u);
    while (!heap.empty())
    {
        diem tg;
        tg=heap.top();heap.pop();
        int u=tg.v,k=tg.k;
        int kk=(k+1)%p;
        for (int i=0;i<dsk[u].size();i++)
        {
            int v=dsk[u][i].first;
            int w=dsk[u][i].second;

```

```

        if (d[v][kk]>d[u][k]+w)
        {
            d[v][kk]=d[u][k]+w;
            diem tg1;
            tg1.v=v;tg1.k=kk;tg1.w=d[v][kk];
            heap.push(tg1);
        }
    }
}
cout<<d[n][0];
}
int main()
{
    nhap();
    khoitao();
    dijkstra();
    return 0;
}

```

### Code pascal

```

program dijkstra;
const fi='dijkstra.inp';
      fo='dijkstra.out';
      gh=10000;
type canh=record
    u,v,c:longint;
end;
var dsc:array[1..gh] of canh;
    dsk,head,gtk,pos,heap,d:array[1..gh*4] of longint;
    free:array[1..gh] of boolean;
    m,n,s,t,nheap:longint;
    f:text;
procedure nhap;

```

```

var i:longint;
begin
    assign(f,fi);reset(f);
    readln(f,n,m);
    for i:=1 to m do
        begin
            readln(f,dsc[i].u,dsc[i].v,dsc[i].c);
            inc(head[dsc[i].u]);
            inc(head[dsc[i].v]);
        end;
    for i:=2 to n do head[i]:=head[i-1]+head[i];
    head[n+1]:=head[n];
    for i:=1 to m do
        begin
            dsk[head[dsc[i].u]]:=dsc[i].v;
            dsk[head[dsc[i].v]]:=dsc[i].u;
            gtk[head[dsc[i].u]]:=dsc[i].c;
            gtk[head[dsc[i].v]]:=dsc[i].c;
            dec(head[dsc[i].u]);
            dec(head[dsc[i].v]);
        end;
    close(f);
end;
procedure khoitao;
var i:longint;
begin
    s:=1;t:=n;
    for i:=1 to n do d[i]:=maxlongint;
    d[s]:=0;
    fillchar(pos,sizeof(pos),0);
    heap:=pos;
    fillchar(free,sizeof(free),true);

```

```

end;
procedure up(v:longint);
var tg,cha:longint;
begin
    cha:=v div 2;
    if (v=1) or (d[heap[cha]]<=d[heap[v]]) then exit;
    tg:=heap[v];heap[v]:=heap[cha];heap[cha]:=tg;
    pos[heap[v]]:=v;pos[heap[cha]]:=cha;
    up(cha);
end;
procedure down(v:longint);
var tg,con:longint;
begin
    con:=v*2;
    if con>nheap then exit;
    if d[heap[con+1]]<d[heap[con]] then con:=con+1;
    if d[heap[v]]<=d[heap[con]] then exit;
    tg:=heap[v];heap[v]:=heap[con];heap[con]:=tg;
    pos[heap[v]]:=v;pos[heap[con]]:=con;
    down(con);
end;
procedure update(v:longint);
var r:longint;
begin
    if pos[v]=0 then
        begin
            inc(nheap);
            heap[nheap]:=v;
            pos[v]:=nheap;
            r:=nheap;
        end
    else r:=pos[v];
    up(r);
end;

```



```

end;
function pop:longint;
begin
    pop:=heap[1];
    heap[1]:=heap[nheap];
    dec(nheap);
    down(1);
end;
procedure dijkstra;
var i,j,u,v:longint;
begin
    update(s);nheap:=1;
    while nheap>0 do
        begin
            u:=pop;
            free[u]:=false;
            for i:=head[u]+1 to head[u+1] do
                begin
                    v:=dsk[i];
                    if free[v] and (d[v]>d[u]+gtk[i]) then
                        begin
                            d[v]:=d[u]+gtk[i];
                            update(v);
                        end;
                end;
            end;
        end;
    end;
end;
procedure hien;
var i:longint;
begin
    assign(f,fo);rewrite(f);
    for i:=1 to n do write(f,d[i], ' ');

```

```

        close(f);
end;
begin
    nhap;
    khoitao;
    dijkstra;
    hien;
end.

```

## Kruskal

### Code c++

```

#include <iostream>

#include <cstdio>

#include <queue>

#include <vector>

using namespace std;

const int gh=100;

struct dsc{int u,v,w;};

bool operator < (dsc x, dsc y)

{

    return x.w>y.w;

}

priority_queue<dsc>heap;

vector<int>root;

vector<dsc> tree;

int n,m,count,sum;

void nhap()

{

    freopen("kruskal.inp","r",stdin);

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

    cin>>n>>m;

    dsc c;

    root.resize(gh);

```

```

    for (int i=0;i<m;i++)
    {
        cin>>c.u>>c.v>>c.w;
        heap.push(c);
    }
}

int getroot(int r)
{
    int i=r;
    while (root[i]>0) i=root[i];
    return i;
}

void hopnhath(int r1,int r2)
{
    int tg=root[r1]+root[r2];
    if (root[r1]>root[r2])
    {
        root[r1]=r2;
        root[r2]=tg;
    } else{
        root[r1]=tg;
        root[r2]=r1;
    }
}

void kruskal()
{
    for (int i=1;i<=n;i++) root[i]=-1;
    for (int i=1;i<=m;i++)
    {
        if (tree.size()==n-1) break;
        dsc c;
        c=heap.top();heap.pop();
    }
}

```

```

    int u=c.u,v=c.v,w=c.w;
    int r1=getroot(u);
    int r2=getroot(v);
    if (r1!=r2)
    {
        tree.push_back(c);
        hopnhat(r1,r2);
    }
}
}
void hien()
{
    for (int i=0;i<tree.size();i++)
        cout<<tree[i].u<<' '<<tree[i].v<<' '<<tree[i].w<<endl;
}
int main()
{
    nhap();
    kruskal();
    hien();
    return 0;
}

```

#### Code pascal

```

program kruskal;
const fi='kruskal.inp';
      fo='kruskal.out';
      gh=10000;
type canh=record
    u,v,c:longint;
end;
var a,t:array[1..gh] of canh;
    root:array[1..gh] of longint;

```

```

    m,n,count,sum:longint;
    f:text;
procedure nhap;
var i:longint;
begin
    assign(f,fi);reset(f);
    readln(f,n,m);
    for i:=1 to m do readln(f,a[i].u,a[i].v,a[i].c);
    close(f);
end;
procedure qsort(l,r:longint);
var i,j:longint;tg,g:canh;
begin
    i:=l;j:=r;g:=a[(i+j) div 2];
    repeat
        while a[i].c<g.c do inc(i);
        while a[j].c>g.c do dec(j);
        if i<=j then
            begin
                tg:=a[i];a[i]:=a[j];a[j]:=tg;
                inc(i);dec(j);
            end;
    until i>j;
    if l<j then qsort(l,j);
    if i<r then qsort(i,r);
end;
function getroot(r:longint):longint;
var i:longint;
begin
    i:=r;
    while root[i]>0 do i:=root[i];
    exit(i);

```

```

end;
procedure hopnhat(r1,r2:longint);
var tg:longint;
begin
    tg:=root[r1]+root[r2];
    if root[r1]<root[r2] then
        begin
            root[r1]:=tg;
            root[r2]:=r1;
        end else
        begin
            root[r1]:=r2;
            root[r2]:=tg;
        end;
    end;
end;
procedure kruskal;
var i,j,u,v,r1,r2:longint;
begin
    for i:=1 to n do root[i]:=-1;
    for i:=1 to m do
        begin
            if count=n-1 then exit;
            u:=a[i].u;v:=a[i].v;
            r1:=getroot(u);
            r2:=getroot(v);
            if r1<>r2 then
                begin
                    inc(count);
                    t[count]:=a[i];
                    sum:=sum+a[i].c;
                    hopnhat(r1,r2);
                end;
        end;
    end;
end;

```

```

        end;
end;
procedure xuli;
var i,j:longint;
begin
    qsort(1,m);
    kruskal;
    assign(f,fo);rewrite(f);
    writeln(f,sum);
    for i:=1 to count do
        writeln(f,t[i].u,' ',t[i].v,' ',t[i].c);
    close(f);
end;
begin
    nhap;
    xuli;
end.

```

### Tazjan

#### Code c++

```

#include <iostream>

#include <cstring>

#include <cstdio>

#include <vector>

#include <stack>

using namespace std;

const int gh=1000;

vector<vector<int> >dsk;

int m,n,count,sl;

vector<int> low,number;

int fre[gh];

stack<int> q;

```

```

void nhap()
{
    freopen("tazan.inp","r",stdin);
    freopen("tazan.out","w",stdout);
    cin>>n>>m;
    dsk.resize(gh);
    for (int i=1;i<=m;i++)
    {
        int u,v;
        cin>>u>>v;
        dsk[u].push_back(v);
        //dsk[v].push_back(u);
    }
    low.resize(gh);number.resize(gh);count=0;sl=0;
}

void dfs(int u)
{
    fre[u]=1;
    q.push(u);
    count++;number[u]=count;low[u]=count;
    for (int i=0;i<dsk[u].size();i++)
    {
        int v=dsk[u][i];
        switch (fre[v])
        {
            case 0: dfs(v);
                    low[u]=min(low[u],low[v]);
                    break;
            case 1: low[u]=min(low[u],number[v]);
                    break;
        }
    }
}

```



```

    }
}
if (low[u]==number[u])
{
    sl++;
    int v;
    cout<<"tplt thu:"<<sl<<endl;
    do
    {
        v=q.top();q.pop();
        cout<<v<<' ';fre[v]=2;
    }while (v!=u);
    cout<<endl;
}
}
void xuli()
{
    memset(fre,0,sizeof(fre));
    for (int i=1;i<=n;i++)
        if (fre[i]==0)
            dfs(i);
}
int main()
{
    nhap();
    xuli();
    return 0;
}

```

**Code pascal**  
program tarjan;  
uses math;

```

const fi='tarjan.inp';
      fo='tarjan.out';
      gh=10000;
type canh=record
      u,v:longint;
      end;
var dsc:array[1..gh] of canh;
      free,dsk,head,number,low,q:array[1..gh] of longint;
      first,m,n,count,sl:longint;
      f:text;
procedure nhap;
var i:longint;
begin
      assign(f,fi);reset(f);
      readln(f,n,m);
      for i:=1 to m do
      begin
            readln(f,dsc[i].u,dsc[i].v);
            inc(head[dsc[i].u]);
      end;
      for i:=2 to n do head[i]:=head[i-1]+head[i];
      head[n+1]:=head[n];
      for i:=1 to m do
      begin
            dsk[head[dsc[i].u]]:=dsc[i].v;
            dec(head[dsc[i].u]);
      end;
      close(f);
      fillchar(free,sizeof(free),0);
end;
procedure dfs(u:longint);
var i,v:longint;

```

```

begin
    free[u]:=1;
    inc(first);q[first]:=u;
    inc(count);number[u]:=count;low[u]:=count;
    for i:=head[u]+1 to head[u+1] do
        begin
            v:=dsk[i];
            case free[v] of
                0:begin
                    dfs(v);
                    low[u]:=min(low[u],low[v]);
                end;
                1:low[u]:=min(low[u],number[v]);
            end;
        end;
    end;
    if low[u]=number[u] then
        begin
            inc(sl);
            writeln(f,'tplt thu ',sl);
            repeat
                v:=q[first];
                write(f,v,' ');
                dec(first);
                free[v]:=2;
            until v=u;
            writeln(f);
        end;
    end;
end;
procedure xuli;
var i:longint;
begin
    assign(f,fo);rewrite(f);

```

```

        for i:=1 to n do if free[i]=0 then dfs(i);
        close(f);
end;
begin
    nhap;
    xuli;
end.

```

### Song liên thông

#### Code pascal

```

program slt;
uses math;
const fi='slt.inp';
      fo='slt.out';
      gh=10000;
type canh=record
    u,v:longint;
end;
var q,dsc:array[1..gh] of canh;
    dsk,head,low,number,pos:array[1..gh] of longint;
    first,id,dem,m,n:longint;
    f:text;
procedure nhap;
var i:longint;
begin
    assign(f,fi);reset(f);
    readln(f,n,m);
    for i:=1 to m do
    begin
        readln(f,dsc[i].u,dsc[i].v);
        inc(head[dsc[i].u]);
        inc(head[dsc[i].v]);
    end;

```

```

for i:=2 to n do head[i]:=head[i]+head[i-1];
head[n+1]:=head[n];
for i:=1 to m do
begin
    dsk[head[dsc[i].u]]:=dsc[i].v;
    dsk[head[dsc[i].v]]:=dsc[i].u;
    pos[head[dsc[i].u]]=head[dsc[i].v];
    pos[head[dsc[i].v]]=head[dsc[i].u];
    dec(head[dsc[i].u]);
    dec(head[dsc[i].v]);
end;
close(f);
end;
procedure dfs(u:longint);
var i,v,x,y:longint;
begin
    inc(id);low[u]:=maxlongint;number[u]:=id;
    for i:=head[u]+1 to head[u+1] do
    begin
        v:=dsk[i];
        if v=0 then continue;
        dsk[pos[v]]=0;
        if number[v]<>0 then low[u]:=min(low[u],number[v])
        else begin
            inc(first);q[first].u:=u;q[first].v:=v;
            dfs(v);
            low[u]:=min(low[u],low[v]);
            if low[v]>=number[u] then
            begin
                inc(dem);
                writeln(f,'tp slt thu:',dem);
                repeat

```

```

        x:=q[first].u;
        y:=q[first].v;
        dec(first);
        write(f,y, ' ');
        until (x=u) and (y=v);
        writeln(f,x);
    end;
end;
end;
end;
procedure xuli;
var i,prev:longint;
begin
    assign(f,fo);rewrite(f);
    for i:=1 to n do
        begin
            prev:=id;
            if number[i]=0 then dfs(i);
            if prev+1=id then
                begin
                    inc(dem);
                    writeln(f,'tp slt thu:',dem);
                    writeln(f,i);
                end;
        end;
    end;
    close(f);
end;
begin
    nhap;
    xuli;
end.

```

## Khớp cầu

### Code pascal

```
program khopcau;

uses math;

const fi='khopcau.inp';
      fo='khopcau.out';
      gh=10000;

type canh=record
    u,v:longint;
end;

var dsc:array[1..gh] of canh;
    free:array[1..gh] of boolean;
    tr,khop,dsk,dsk1,head,pos,low,number:array[1..gh] of longint;
    id,dem,m,n:longint;
    f:text;

procedure nhap;
var i:longint;
begin
    assign(f,fi);reset(f);
    readln(f,n,m);
    for i:=1 to m do
    begin
        readln(f,dsc[i].u,dsc[i].v);
        inc(head[dsc[i].u]);
        inc(head[dsc[i].v]);
    end;
    for i:=2 to n do head[i]:=head[i]+head[i-1];
    head[n+1]:=head[n];
    for i:=1 to m do
    begin
        dsk[head[dsc[i].u]]:=dsc[i].v;
        dsk[head[dsc[i].v]]:=dsc[i].u;
```

```

        pos[head[dsc[i].u]]:=head[dsc[i].v];
        pos[head[dsc[i].v]]:=head[dsc[i].u];
        dec(head[dsc[i].u]);
        dec(head[dsc[i].v]);
    end;
    dsk1:=dsk;
    close(f);
    fillchar(free,sizeof(free),true);
end;

procedure dfs(u:longint);
var i,v:longint;
begin
    free[u]:=false;
    inc(id);low[u]:=id;number[u]:=id;
    for i:=head[u]+1 to head[u+1] do
        begin
            v:=dsk[i];
            if v=0 then continue;
            dsk[pos[i]]:=0;
            if free[v] then
                begin
                    tr[v]:=u;
                    dfs(v);
                    low[u]:=min(low[u],low[v]);
                end else low[u]:=min(low[u],number[v]);
            end;
        end;
end;

procedure timkhop;
var i,j,u,v,t1,t2:longint;
begin
    dsk:=dsk1;
    for i:=1 to n do

```



```

begin
    u:=i;t1:=0;t2:=0;
    for j:=head[u]+1 to head[u+1] do
        begin
            v:=dsk[j];
            if tr[v]=u then
                begin
                    inc(t1);
                    if low[v]>=number[u] then inc(t2);
                end;
            if tr[u]=0 then khop[u]:=t1-1
            else khop[u]:=t2;
        end;
    end;
    j:=0;
    for i:=1 to n do if khop[i]<>0 then inc(j);
    writeln(f,'so khop la:',j);
end;

procedure timcau;
var i,j,u,v:longint;
begin
    j:=0;
    for i:=1 to m do
        begin
            u:=dsc[i].u;v:=dsc[i].v;
            if (tr[v]=u) and (low[v]>number[u]) then inc(j);
            if (tr[u]=v) and (low[u]>number[v]) then inc(j);
        end;
    writeln(f,'so cau la:',j);
end;

procedure xuli;
var i:longint;

```

```

begin
    assign(f,fo);rewrite(f);
    for i:=1 to n do if free[i] then dfs(i);
    timkhop;timcau;
    close(f);
end;
begin
    nhap;
    xuli;
end.

```

## Stack

### Codepascal

```

program stack;
const fi='stack.inp';
    fo='stack.out';
    gh=10000;
var a,s,prev,next:array[0..gh] of longint;
    sn,n:longint;
    f:text;
procedure nhap;
var i:longint;
begin
    assign(f,fi);reset(f);
    readln(f,n);
    for i:=1 to n do read(f,a[i]);
    close(f);
end;
procedure xuli;
var i:longint;
begin
    sn:=0;a[0]:=-maxlongint;
    for i:=1 to n do

```

```

if a[i]>a[i-1] then
begin
    prev[i]:=i;
    inc(sn);s[sn]:=i;
end else
begin
    while (a[s[sn]]>=a[i]) and (sn>0) do dec(sn);
    inc(sn);prev[i]:=prev[s[sn]];
    s[sn]:=i;
end;
sn:=0;a[n+1]:=-maxlongint;
for i:=n downto 1 do
if a[i]>a[i+1] then
begin
    next[i]:=i;
    inc(sn);s[sn]:=i;
end else
begin
    while (a[s[sn]]>=a[i]) and (sn>0) do dec(sn);
    inc(sn);next[i]:=next[s[sn]];
    s[sn]:=i;
end;
end;
procedure hien;
var i:longint;
begin
    assign(f,fo);rewrite(f);
    for i:=1 to n do write(f,prev[i],' ');writeln(f);
    for i:=1 to n do write(f,next[i],' ');
    close(f);
end;
begin

```

```

        nhap;
        xuli;
        hien;
end.

```

## Hash

### Code c++

```

#include <iostream>

#include <cstdio>

using namespace std;

const long long gh=2000000;

const long long base=1000007;

long long pow[gh],hasht[gh],n,m,hashp;

string p,t;

void nhap()
{
    ios_base::sync_with_stdio(0);
    cin>>t;cin>>p;
}

void khoitao()
{
    pow[0]=1;m=t.size();n=p.size();
    for (int i=1;i<=m;i++) pow[i]=(pow[i-1]*26) % base;
    for (int i=1;i<=m;i++) hasht[i]=(hasht[i-1]*26+t[i-1]-'a') % base;
    for (int i=1;i<=n;i++) hashp=(hashp*26+p[i-1]-'a') % base;
}

long long gethasht(long long i,long long j)
{
    long long tg= (( hasht[j]-hasht[i-1]*pow[j-i+1]+base*base )%base);
    return tg;
}

void xuli()

```

```

{
    khoitao();
    for (int i=1;i<=m-n+1;i++)
        if (hashp==gethasht(i,i+n-1))
            cout<<i<<' ';
}
int main()
{
    nhap();
    xuli();
    return 0;
}

```

#### Code pascal

```

program hash;
const fi='hash.inp';
      fo='hash.out';
      gh=100000;
      base=10000007;
var t,p:ansistring;
    hashp,m,n:longint;
    pow,hasht:array[0..gh] of int64;
    f:text;
procedure nhap;
begin
    assign(f,fi);reset(f);
    readln(f,t);readln(f,p);
    close(f);
end;
procedure khoitao;
var i:longint;

```

```

begin
    m:=length(t);n:=length(p);
    pow[0]:=1;
    for i:=1 to m do pow[i]:=pow[i-1]*26 mod base;
    hasht[0]:=0;
    for i:=1 to m do
        hasht[i]:=(hasht[i-1]*26 + ord(t[i]) - ord('a')) mod base;
    hashp:=0;
    for i:=1 to n do
        hashp:=(hashp*26 + ord(p[i]) - ord('a')) mod base;
    end;
function gethash(i,j:int64):int64;
begin
    gethash:=(hasht[j]-hasht[i-1]*pow[j-i+1] +base*base) mod base;
end;
procedure xuli;
var i:longint;
begin
    assign(f,fo);rewrite(f);
    for i:=1 to m-n+1 do
        if gethash(i,i+n-1)=hashp then write(f,i, ' ');
    close(f);
end;
begin
    nhap;
    khoitao;
    xuli;
end.

```

## Deque

Code pascal

```

program dquece;
const fi='dquece.inp';

```

```

fo='dqueuece.out';
gh=100000;
var a,i0,j0,q,ds:array[0..gh] of longint;
n,m:longint;
f:text;
procedure nhap;
var i:longint;
begin
    assign(f,fi);reset(f);
    readln(f,n,m);
    for i:=1 to n do read(f,a[i]);
    for i:=1 to m do
        readln(f,i0[i],j0[i]);
    close(f);
end;
procedure xuli;
var i,j,l,r,u,v:longint;
begin
    i0[0]:=0;j0[0]:=0;
    l:=1;r:=0;
    for u:=1 to n do
        begin
            for v:=j0[u-1]+1 to j0[u] do
                begin
                    while (l<=r) and (q[r]>a[v]) do dec(r);
                    inc(r);q[r]:=a[v];
                end;
            for v:=i0[u-1] to i0[u]-1 do
                if (a[v]=q[l]) then inc(l);
            ds[u]:=q[l];
        end;
    end;
end;

```

```

procedure hien;
var i:longint;
begin
    assign(f,fo);rewrite(f);
    for i:=1 to m do
        writeln(f,ds[i]);
    close(f);
end;
begin
    nhap;
    xuli;
    hien;
end.

```

### Euler

```

#include <bits/stdc++.h>

using namespace std;

int n,m,w,a[300][300];
stack <int> s;

int main()
{
    scanf("%d%d",&n,&m);
    for (int i=1;i<=n;i++) scanf("%d",&w);
    for (int i=1;i<=m;i++) {
        int u,v;
        scanf("%d%d",&u,&v);
        a[u][v]++;
        a[v][u]++;
    }
    s.push(1);

```



```

printf("%d\n",m);
while (not s.empty()) {
    int u = s.top();
    bool kt = true;
    for (int v=1;v<=n;v++) {
        if (a[u][v] > 0) {
            a[u][v]--;
            a[v][u]--;
            kt = false;
            s.push(v);
            break;
        }
    }
    if (kt) {
        printf("%d ",u);
        s.pop();
    }
}
return 0;
}

```

### Floyd

```

#include <iostream>
#include <cstdio>
using namespace std;
const int oo = 1000111000;
int a[239][239];
int n, m,s,t;

void minimize(int &a, int b){ if (a>b) a=b; }

main(){
    freopen("floyd.inp","r",stdin);

```

```

freopen("floyd.out","w",stdout);
int i,j,k, p,q,w;
cin>>n>>m>>s>>t;
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
a[i][j] = oo;
for (i=1; i<=n; i++)
a[i][i] = 0;
for (i=1; i<=m; i++)
{
    cin>>p>>q>>w;
    a[p][q] = a[q][p] = w;
}

for (k=1; k<=n; k++)
for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
    minimize(a[i][j], a[i][k] + a[k][j]);
for (int i=1;i<=n;i++) cout<<a[1][i]<<' ';
return 0;
}

```