```
vector<int> adj[2100];
vector<int> radj[2100];
inline int TRUE(int x) {return x + x;}
inline int FALSE(int x) {return x + x + 1;}
inline void link(int x,int y) {adj[x].push_back(y);radj[y].push_back(x);}
int n,m;
vector<int> st;
bool visit[2100];
int fa[2100];

void dfs_1(int now) {
    if (visit[now]) return;
    visit[now] = true;
    for (int i = 0; i < adj[now].size(); i++) {
        int next = adj[now][i];
        dfs_1(next);
    }
    st.push_back(now);
}
void dfs_2(int now,int blk) {
    if (visit[now]) return;
    visit[now] = true;
    fa[now] = blk;
    for (int i = 0; i < radj[now].size(); i++) {
        int next = radj[now][i];
        dfs_2(next,blk);
    }
}

bool check() {
    memset(visit,0,sizeof(visit));
    for (int i = 0; i < (n << 1); i++) {
        if (!visit[i]) {
            dfs_1(i);
        }
    }
    memset(visit,0,sizeof(visit));
    memset(fa,0xff,sizeof(fa));
    int nn = 0;
    while (st.size()) {
        int now = *st.rbegin(); st.pop_back();
        if (!visit[now]) {
            dfs_2(now, ++nn);
        }
    }
    for (int i = 0; i < n; i++) {
        if (fa[TRUE(i)] == fa[FALSE(i)]) {
            return false;
        }
    }
    return true;
}
```

```
//FFT+高精度
typedef long long Long;
const int MAXN=32768;
const double pi=acos(-1.0);
const Long MOD=100000;
const int TEN=5;

double ra[MAXN];
double ia[MAXN];
double rb[MAXN];
double ib[MAXN];
double rc[MAXN];
double ic[MAXN];
char a[MAXN];
char b[MAXN];
int slena;
int slenb;
int lena;
int lenb;
int n,logn;
Long ans[MAXN];
double R[MAXN];
double I[MAXN];

int rev(int x,int bit)
{
    int ans=0;
    for (int i=0;i<bit;i++)
    {
        ans<<=1;
        if (x&1) ans|=1;
        x>>=1;
    }
    return ans;
}
```

```
void fft(double ir[],double ii[],int size,int mark)
{
    double delta=mark*2*pi;
    for (int i=0;i<size;i++)
    {
        int tt=rev(i,logn);
        R[tt]=ir[i];
        I[tt]=ii[i];
    }
    for (int s=1;s<=logn;s++)
    {
        int m=1<<s;
        double rwm=cos(delta/m);
        double iwm=sin(delta/m);
        for (int k=0;k<n;k+=m)
        {
            double rw=1;
            double iw=0;
            for (int j=0;j<m/2;j++)
            {
                double rt=rw*R[k+j+m/2]-iw*I[k+j+m/2];
                double it=rw*I[k+j+m/2]+iw*R[k+j+m/2];
                double ru=R[k+j];
                double iu=I[k+j];
                R[k+j]=ru+rt;
                I[k+j]=iu+it;
                R[k+j+m/2]=ru-rt;
                I[k+j+m/2]=iu-it;
                double rnw=rw*rwm-iw*iwm;
                double inw=rw*iwm+iw*rwm;
                rw=rnw; iw=inw;
            }
        }
    }
    for (int i=0;i<size;i++)
    {
        ir[i]=R[i];
        ii[i]=I[i];
    }
}
```

```
double
POW[10]={1,10,100,1000,10000,100000,1000000,10000000,100000000,1000000000};

int next(char str[])
{
     int len=0;
     for (str[len]=getchar();str[len]>='0';str[len]=getchar())
          len++;
     str[len]=0;
     return len;
}
int main()
{
     int nn=0;
     scanf("%d",&nn); getchar();
     while (nn--)
     {
          memset(ra,0,n<<3);
          memset(ia,0,n<<3);
          memset(rb,0,n<<3);
          memset(ib,0,n<<3);
          memset(ans,0,n<<3);

          slena=next(a);
          int cnt=0; lena=0;
          for (int j=slena-1;j>=0;j--)
          {
               ra[lena]=ra[lena]+(a[j]-'0')*POW[cnt++];
               if (cnt==TEN) {lena++; cnt=0;}
          }
          if (ra[lena]>0.1)    lena++;

          slenb=next(b);
          cnt=0; lenb=0;
          for (int j=slenb-1;j>=0;j--)
          {
               rb[lenb]=rb[lenb]+(b[j]-'0')*POW[cnt++];
               if (cnt==TEN) {lenb++; cnt=0;}
          }
          if (rb[lenb]>0.1)    lenb++;
```

```
        n=1; logn=0;
        while (n<lena || n<lenb) {n+=n;logn++;}
        n+=n; logn++;

        fft(ra,ia,n,1);
        fft(rb,ib,n,1);
        for (int i=0;i<n;i++)
        {
            rc[i]=ra[i]*rb[i]-ia[i]*ib[i];
            ic[i]=ra[i]*ib[i]+rb[i]*ia[i];
        }
        fft(rc,ic,n,-1);
        for (int i=0;i<n;i++)
            ans[i]=(Long)(rc[i]/n+0.5);
        for (int i=0;i<n-1;i++)
        {
            ans[i+1]+=ans[i]/MOD;
            ans[i]%=MOD;
        }
        bool print=0;
        for (int i=n-1;i>=0;i--)
        {
            if (!print && (ans[i]>0 || i==0))
            {
                print=1;
                printf("%lld",ans[i]);
            } else
            if (print)
                printf("%05lld",ans[i]);
        }
        putchar(10);
    }
    return 0;
}
```

```
const int MAXN = 210;
const int MAXM = 500010;
const int inf = 2E9;

typedef struct {int v,next,val;} edge;
struct SAP {
    edge e[MAXM];
    int p[MAXN],eid;
    inline void clear(){memset(p,-1,sizeof(p));eid=0;}
    inline void insert1(int from,int to,int val) {
        e[eid].v=to;
        e[eid].val=val;
        e[eid].next=p[from];
        p[from]=eid++;
        swap(from,to);
        e[eid].v=to;
        e[eid].val=0;
        e[eid].next=p[from];
        p[from]=eid++;
    }
    inline void insert2(int from,int to,int val) {
        e[eid].v=to;
        e[eid].val=val;
        e[eid].next=p[from];
        p[from]=eid++;
        swap(from,to);
        e[eid].v=to;
        e[eid].val=val;
        e[eid].next=p[from];
        p[from]=eid++;
    }
    int n;//为点数n 为边数m
    int h[MAXN];
    int gap[MAXN];
    int source,sink;
    inline int dfs(int pos,int cost) {
        if (pos==sink) {
            return cost;
        }
        int j,minh=n-1,lv=cost,d;
        for (j=p[pos];j!=-1;j=e[j].next) {
            int v=e[j].v,val=e[j].val;
            if(val>0) {
                if (h[v]+1==h[pos]) {
                    if (lv<e[j].val) d=lv;
                    else d=e[j].val;
                    d=dfs(v,d);
                    e[j].val-=d;
                    e[j^1].val+=d;
                    lv-=d;
                    if (h[source]>=n) return cost-lv;
                    if (lv==0) break;
                }
                if (h[v]<minh)  minh=h[v];
            }
        }
        if (lv==cost) {
            --gap[h[pos]];
            if (gap[h[pos]]==0) h[source]=n;
            h[pos]=minh+1;
            ++gap[h[pos]];
        }
        return cost-lv;
    }
    void read(int ll[MAXN][MAXN],int S,int T,int N) {
        clear();
        source = S; sink = T; n = N;
        for (int i = 0; i <= N; i++) {
            for (int j = 0; j <= N; j++) {
                if (ll[i][j]) {
                    insert1(i,j,ll[i][j]);
                }
            }
        }
    }
    int run() {
        int ret=0;
        memset(gap,0,sizeof(gap));
        memset(h,0,sizeof(h));
        gap[source]=n;
        while (h[source]<n) ret+=dfs(source,inf);
        return ret;
    }
} solver;
```

```java
import java.util.concurrent.CyclicBarrier;
import java.util.*;
import java.io.*;
import java.math.*;

class Main {
    public static String[] ans;
    void run() {
        ArrayList<SubTask> tasks = new ArrayList<SubTask>();
        // read the input data for each task...
        // while (!EOF) tasks.add(new SubTask().read());
        int ts = tasks.size();
        // ans = new String[ts];
        // CyclicBarrier cb = new CyclicBarrier(ts, new MainTask());
        // set cb and pos for all subtask and run them....
    }
    public static void main(String[] args) {
        new Main().run();
    }
}

class MainTask implements Runnable {
    public void run() {
        PrintWriter out = new PrintWriter(System.out);
        for (String x : Main.ans) {
            out.println(x);
        }
        out.flush();
    }
}

class SubTask extends Thread {
    int pos;
    CyclicBarrier cb;

//  input data for each case.....
    void read() {
        //read it..
    }

    public void run() {
        // solve it....
        // Main.ans[pos] = ans;
        try { cb.await(); } catch (Exception e) { }
    }
}
```

```
import java.util.*;
import java.io.*;
import java.math.*;


class Main {

    void solve() throws Exception {
        MyReader in = new MyReader();
        //.
    }

    public static void main(String args[]) throws Exception {
        new Main().solve();
    }

    void debug(Object...x) {
        System.out.println(Arrays.deepToString(x));
    }
}

class MyReader {
    BufferedReader br = new BufferedReader (
            new InputStreamReader (System.in));
    StringTokenizer in;
    String next() throws Exception {
        if (in == null || !in.hasMoreTokens()) {
            in = new StringTokenizer(br.readLine());
        }
        return in.nextToken();
    }
    int nextInt() throws Exception {
        return Integer.parseInt(next());
    }
}
```

```
#include <cstdio>
#include <cstring>
using namespace std;

const int maxn=160,OO=2147483647;
int w[maxn][maxn];
int lx[maxn],ly[maxn];
int linky[maxn];
int visx[maxn],visy[maxn];
int N;
int slack[maxn];

void input(){
    scanf("%d",&N);
    for(int i=0;i<N;++i)
        for(int j=0;j<N;++j)
            scanf("%d",&w[i][j]);
}
bool find(int x){
    visx[x]=true;
    for(int y=0;y<N;++y){
        if(visy[y])continue;
        int t=lx[x]+ly[y]-w[x][y];
        if(t==0){
            visy[y]=true;
            if(linky[y]==-1||find(linky[y])){
                linky[y]=x;
                return true;
            }
        }
        else{
            if(slack[y]>t)
                slack[y]=t;
        }
    }
    return false;
}
void KM(){
    memset(linky,-1,sizeof(linky));
    memset(lx,0,sizeof(lx));
    memset(ly,0,sizeof(ly));
    for(int i=0;i<N;++i)
        for(int j=0;j<N;++j)
            if(w[i][j]>lx[i])
                lx[i]=w[i][j];
    for(int x=0;x<N;++x){
        for(int i=0;i<N;++i)
            slack[i]=OO;
        for(;;){
            memset(visx,0,sizeof(visx));
            memset(visy,0,sizeof(visy));
            if(find(x))break;
            int d=OO;
            for(int i=0;i<N;++i){
                if(!visy[i])
                    if(d>slack[i])
                        d=slack[i];
            }
            for(int i=0;i<N;++i){
                if(visx[i])
                    lx[i]-=d;
            }
            for(int i=0;i<N;++i){
                if(visy[i])
                    ly[i]+=d;
                else
                    slack[i]-=d;
            }
        }
    }
}
void output(){
    int res=0;
    for(int j=0;j<N;++j){
        for(int i=0;i<N;++i)
            res+=w[i][j];
        res-=w[linky[j]][j];
    }
    printf("%d\n",res);
}
int main(){
    input();
    KM();
    output();
}
```

```
#include <cstdio>
#include <vector>
#include <cstring>

using namespace std;

int deep[80010];
int cnt;
int pos[40010];
bool visit[40010];
vector<int> adj[40010];
vector<int> cost[40010];
int n,m;
char buf[100];
int data[320010];

void dfs(int now,int len) {
    deep[cnt] = len;
    pos[now] = cnt;
    cnt ++;
    visit[now] = true;
    for (int i = 0; i < adj[now].size(); i++) {
        int t = adj[now][i];
        int v = cost[now][i];
        if (!visit[t]) {
            dfs(t,len + v);
            deep[cnt] = len;
            pos[now] = cnt;
            cnt ++;
        }
    }
}

void init(int now,int left,int right) {
    if (left == right) { data[now] = deep[left]; return; }
    int mid = (left + right) >> 1;
    init(now + now, left, mid);
    init(now + now + 1, mid + 1, right);
    data[now] = data[now + now];
    if (data[now + now + 1] < data[now])
        data[now] = data[now + now + 1];
}

int query(int now,int left,int right,int l,int r) {
    if (l <= left && right <= r) return data[now];
    if (l > right || r < left) return 0x7fffffff;
    int mid = (left + right) >> 1;
    int tl = query(now + now,left,mid,l,r);
    int tr = query(now + now + 1,mid + 1,right,l,r);
    return tl > tr ? tr : tl;
}

```

```
int main() {
    scanf("%d%d",&n,&m);
    for (int i = 0; i < m; i++) {
        int f,t,v;
        scanf("%d%d%d",&f,&t,&v);
        scanf("%s",buf);
        adj[f].push_back(t);
        cost[f].push_back(v);
        adj[t].push_back(f);
        cost[t].push_back(v);
    }
    dfs(1,0); init(1,0,cnt - 1);
    int k; scanf("%d",&k);
    for (int i = 0; i < k; i++) {
        int f,t;
        scanf("%d%d",&f,&t);
        f = pos[f]; t = pos[t];
        if (f > t) {int tmp = f; f = t; t = tmp;}
        int LCA = query(1, 0, cnt - 1, f, t);
        printf("%d\n",deep[f] + deep[t] - LCA * 2);
    }
    return 0;
}
```

POJ3352，边双联通分量

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

using namespace std;

int a[1010][1010];
int visit[1010];
int deep[1010];
int back[1010];
int n,m;

int block[1010];
int dd[1010];

void dfs(int k,int fa,int d) {
    visit[k]=1;
    back[k]=deep[k]=d;
    int tot=0;
    for (int i=1;i<=n;i++) {
        if (a[k][i] && i!=fa && visit[i]==1)
            back[k]=min(back[k],deep[i]);
        if (a[k][i] && visit[i]==0) {
            dfs(i,k,d+1);
            tot++;
            back[k]=min(back[k],back[i]);
            if (back[i]>deep[k]) {
                a[k][i] = a[i][k] = 2;
            }
        }
    }
    visit[k]=2;
}

void dfs2(int k,int fa) {
    visit[k]=1; block[k] = fa;
    for (int i=1;i<=n;i++) {
        if (a[k][i] == 1 && !visit[i]) {
            dfs2(i,fa);
        }
    }
}

int main() {
    scanf("%d%d",&n,&m);
    memset(a,0,sizeof(a));
    memset(back,0,sizeof(back));
    memset(deep,0,sizeof(deep));
    memset(visit,0,sizeof(visit));
    while (m--) {
        int f,t;
        scanf("%d%d",&f,&t);
        a[f][t]=a[t][f]=1;
    }
    dfs(1,0,0);
    memset(visit,0,sizeof(visit));
    for (int i = 1; i <= n; i++) {
        if (!visit[i])
            dfs2(i,i);
    }
    for (int i = 1; i <= n; i++) {
        for (int j = i; j <= n; j++) {
            if (a[i][j] == 2) {
                dd[block[i]]++;
                dd[block[j]] ++;
            }
        }
    }
    int cnt = 1;
    for (int i = 1; i <= n; i++) if (dd[i] == 1) cnt ++;
    printf("%d\n",cnt >> 1);
    return 0;
}
```

```
//AC 自动机
struct trie{
    trie *next[4];
    trie *fail;
    bool isend;
};
void insert(char s[]) {
    trie *now=root;
    for (;;) {
        if (s[0]==0) {
            now->isend=1;
            return;
        }
        int tt=s[0]-'0';
        if (now->next[tt]==NULL) now->next[tt]=++head;
        now=now->next[tt];
        s++;
    }
}
void buildFaliure() {
    queue<trie*> q;
    for (int i=0;i<4;i++)
    if (root->next[i]) {
        root->next[i]->fail=root;
        q.push(root->next[i]);
    } else root->next[i]=root;
    while (!q.empty()) {
        trie *now=q.front(); q.pop();
        for (int i=0;i<4;i++) {
            trie *u=now->next[i];
            if (u) {
                q.push(u);
                trie *v=now->fail;
                while (v->next[i]==NULL)
                    v=v->fail;
                u->fail=v->next[i];
            }
        }
        if (now->fail->isend) now->isend=1;
    }
}
trie* go(trie *now,char ch) {
    ch-='0';
    trie *ans=now;
    while (ans->next[ch]==NULL)
        ans=ans->fail;
    return ans->next[ch];
}
```

```
//nlogn Dijkstra
struct node{
    int dist,n;
    node(int x,int y){
        n=x; dist=y;
    }
    bool operator < (const node &t) const {
        return dist>t.dist;
    }
};

int a[1010][2000];
int b[1010][2000];
int dist[1010];

int main(){
    int n,m;
    scanf("%d%d",&m,&n);
    while (m--){
        int f,t,cost;
        scanf("%d%d%d",&f,&t,&cost);
        a[f][++a[f][0]]=t;
        b[f][++b[f][0]]=cost;
        a[t][++a[t][0]]=f;
        b[t][++b[t][0]]=cost;
    }
    memset(dist,63,sizeof(dist));
    priority_queue<node> q;
    dist[n]=0; q.push(node(n,0));
    while (!q.empty()&&!visit[1]){
        int v=q.top().n;
        int d=q.top().dist;
        q.pop();
        if (d<=dist[v]) {
            for (int i=1;i<=a[v][0];i++)
            if (dist[a[v][i]]>dist[v]+b[v][i]){
                dist[a[v][i]]=dist[v]+b[v][i];
                q.push(node(a[v][i],dist[a[v][i]]));
            }
        }
    }
    printf("%d\n",dist[1]);
    return 0;
}
```

```
//KMP
void init(char s[],int next[],int n)
{
        next[0]=next[1]=0;
        for (int i=2;i<=n;i++)
        {
                int j=next[i-1];
                while (j>0)
                {
                        if (s[j]==s[i-1]) break;
                        j=next[j];
                }
                if (s[j]==s[i-1]) j++;
                next[i]=j;
        }
}
int main()
{
        int l;
        while (scanf("%d",&l)!=EOF)
        {
                char *s=new char[l+10];
                scanf("%s",s);

                int *next=new int[l+10];
                init(s,next,l);
                getchar();
                int j=0; int pos=0;
                bool ans=0;
                for (;;)
                {
                        char c=getchar();
                        if (c==10) break;
                        while (j>0 && s[j]!=c) j=next[j];
                        if (s[j]==c) j++;
                        if (j==l)
                        {
                                printf("%d\n",pos-l+1);
                                ans=1;
                                j=next[j];
                        }
                        pos++;
                }
                if (!ans) printf("\n");
                delete []s;
                delete []next;
        }
        return 0;
}
```

```
//SA
#define maxn 1000001
int wa[maxn],wb[maxn],wv[maxn],ws[maxn];
int cmp(int *r,int a,int b,int l)
{return r[a]==r[b]&&r[a+l]==r[b+l];}
void da(int *r,int *sa,int n,int m)//n+1,m:字符集大小，sa：[1,N]
{
     int i,j,p,*x=wa,*y=wb,*t;
     for(i=0;i<m;i++) ws[i]=0;
     for(i=0;i<n;i++) ws[x[i]=r[i]]++;
     for(i=1;i<m;i++) ws[i]+=ws[i-1];
     for(i=n-1;i>=0;i--) sa[--ws[x[i]]]=i;
     for(j=1,p=1;p<n;j*=2,m=p)
     {
       for(p=0,i=n-j;i<n;i++) y[p++]=i;
       for(i=0;i<n;i++) if(sa[i]>=j) y[p++]=sa[i]-j;
       for(i=0;i<n;i++) wv[i]=x[y[i]];
       for(i=0;i<m;i++) ws[i]=0;
       for(i=0;i<n;i++) ws[wv[i]]++;
       for(i=1;i<m;i++) ws[i]+=ws[i-1];
       for(i=n-1;i>=0;i--) sa[--ws[wv[i]]]=y[i];
       for(t=x,x=y,y=t,p=1,x[sa[0]]=0,i=1;i<n;i++)
       x[sa[i]]=cmp(y,sa[i-1],sa[i],j)?p-1:p++;
     }
     return;
}
int rank[maxn],height[maxn];
void calheight(int *r,int *sa,int n)
{
     int i,j,k=0;
     for(i=1;i<=n;i++) rank[sa[i]]=i;
     for(i=0;i<n;height[rank[i++]]=k)
     for(k?k--:0,j=sa[rank[i]-1];r[i+k]==r[j+k];k++);
     return;
}
```

```
int RMQ[maxn];
int mm[maxn];
int best[20][maxn];
void initRMQ(int n)
{
     int i,j,a,b;
     for(mm[0]=-1,i=1;i<=n;i++)
     mm[i]=((i&(i-1))==0)?mm[i-1]+1:mm[i-1];
     for(i=1;i<=n;i++) best[0][i]=i;
     for(i=1;i<=mm[n];i++)
     for(j=1;j<=n+1-(1<<i);j++)
     {
       a=best[i-1][j];
       b=best[i-1][j+(1<<(i-1))];
       if(RMQ[a]<RMQ[b]) best[i][j]=a;
       else best[i][j]=b;
     }
     return;
}
int askRMQ(int a,int b)
{
    int t;
    t=mm[b-a+1];b-=(1<<t)-1;
    a=best[t][a];b=best[t][b];
    return RMQ[a]<RMQ[b]?a:b;
}
int lcp(int a,int b)
{
    int t;
    a=rank[a];b=rank[b];
    if(a>b) {t=a;a=b;b=t;}
    return(height[askRMQ(a+1,b)]);
}
```

From：POJ1144，割点、割边

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

using namespace std;

bool a[110][110];
int visit[110];
int deep[110];
int back[110];
bool cut[110];
int n,ans;


void dfs(int k,int fa,int d)
{
    visit[k]=1;
    back[k]=deep[k]=d;
    int tot=0;
    for (int i=1;i<=n;i++)
    {
        if (a[k][i] && i!=fa && visit[i]==1)
            back[k]=min(back[k],deep[i]);
        if (a[k][i] && visit[i]==0)
        {
            dfs(i,k,d+1);
            tot++;
            back[k]=min(back[k],back[i]);
            if ((k==1 &&tot>1) || (k!=1 && back[i]>=deep[k]))
                if (!cut[k])
                {
                    cut[k]=1;
                    ans++;
                }
          //if back[i]>deep[k] k,i is bridge;
        }
    }
    visit[k]=2;
}
```

```
int main()
{
    while (1)
    {
        scanf("%d",&n);
        if (n==0)
            break;
        memset(a,0,sizeof(a));
        memset(back,0,sizeof(back));
        memset(cut,0,sizeof(cut));
        memset(deep,0,sizeof(deep));
        memset(visit,0,sizeof(visit));
        ans=0;
        int f;
        while (scanf("%d",&f) && f>0)
        {
            while (getchar()!=10)
            {
                int t;
                scanf("%d",&t);
                a[f][t]=a[t][f]=1;
            }
        }
        dfs(1,0,0);
        printf("%d\n",ans);
    }
    return 0;
}
```

From:POJ3041，二分图

```
#include <cstdio>
#include <cstring>

bool a[1010][1010];
bool visit[1010];
int match[1010];
int n;

bool dfs(int k)
{
      for (int i=1;i<=n+n;i++)
      if (!visit[i]&&a[k][i])
      {
            visit[i]=1;
            int tt=match[i];
            match[i]=k;
            if (tt==0||dfs(tt)) return 1;
            match[i]=tt;
      }
      return 0;
}

int main()
{
    int m;
    scanf("%d%d",&n,&m);
    while (m--)
    {
        int x,y;
        scanf("%d%d",&x,&y);
        a[x][y+n]=a[y+n][x]=1;
    }
    int ans=0;
    for (int i=1;i<=n;i++)
    {
        memset(visit,0,sizeof(visit));
        if (dfs(i))
            ans++;
    }
    printf("%d\n",ans);
    return 0;
}
```

From：POJ2299，逆序对

```
#include <cstdio>

int a[500010];
int t[500010];
long long ans;

void merge(int a[],int sizea,int b[],int sizeb)
{
    int nowa=0;
    int nowb=0;
    int s=0;
    while (nowa<sizea&&nowb<sizeb)
    {
        if (a[nowa]<=b[nowb])
            t[s++]=a[nowa++];
        else
        if (a[nowa]>b[nowb])
        {
            t[s++]=b[nowb++];
            ans+=sizea-nowa;
        }
    }
    while (nowa<sizea)
        t[s++]=a[nowa++];
    while (nowb<sizeb)
        t[s++]=b[nowb++];
}

void sort(int a[],int size)
{
    if (size<2)
        return;
    int lsize=size>>1;
    int rsize=size-lsize;
    sort(a,lsize);
    sort(a+lsize,rsize);
    merge(a,lsize,a+lsize,rsize);
    for (int i=0;i<size;i++)
        a[i]=t[i];
}
```

杂

```
void gcd(int a,int b, int &d, int &x, int &y)
{
      if (b==0)
      {
            x=1;
            y=0;
            d=a;
      }
      else
      {
            int x1,y1;
            gcd(b,a%b,d,x1,y1);
            x=y1;
            y=x1-(a/b)*y1;
      }
}

int elfhash(char *key)
{
    unsigned long h=0;
    while(*key)
    {
        h=(h<<4)+*key++;
        unsigned long g=h&0Xf0000000L;
        if(g) h^=g>>24;
        h&=~g;
    }
    return h%MOD;
}
```

BIT:

```
int sum(int k)
{
    int ans = 0;
    for (int i=k;i>0;i-=i&-i)
        ans += a[i];
    return ans;
}

void change(int k,int n,int delta)
{
    for (int i=k;i<=n;i+=i&-i)//小心i=0死循环
        a[i]+=delta;
}
```

```
//POJ2195 新最小费用流
int n,m,ans,t,f;
int maxf[210][210],flow[210][210],dist[210][210];
int fa[210],cost[210];
bool inque[210];
inline int abs(int a) {return a>0?a:-a;}
void init()
{
      int a[210][2]={0},b[210][2]={0},s=0,sa=0,sb=0;
      memset(maxf,0,sizeof(maxf));
      memset(flow,0,sizeof(flow));
      memset(dist,0,sizeof(dist));
      for (int i=1;i<=n;i++)
      for (int j=1;j<=m;j++)
      {
            char tt;
            cin>>tt;
            if (tt=='H')
            {
                  a[++sa][0]=i;
                  a[sa][1]=j;
            }
            if (tt=='m')
            {
                  b[++sb][0]=i;
                  b[sb][1]=j;
            }
      }
      s=sa;
      for (int i=1;i<=s;i++)
      for (int j=1;j<=s;j++)
      {
            dist[i][s+j]=abs(a[i][0]-b[j][0])+abs(a[i][1]-b[j][1]);
            dist[s+j][i]=dist[i][s+j];
            maxf[i][s+j]=1;
      }
      for (int i=1;i<=s;i++)
            maxf[0][i]=maxf[s+i][s+s+1]=1;
      t=s+s+1;
      f=0;
      ans=0;
}

inline int value(int i,int j){
      return flow[j][i]>0?-dist[i][j]:dist[i][j];
}
```

```
bool spfamark()
{
      memset(fa,0,sizeof(fa));
      memset(inque,0,sizeof(inque));
      for (int i=1;i<=t;i++)
            cost[i]=2000000000;
      queue<int> q;
      q.push(f); inque[f]=1; cost[f]=0;
      while (!q.empty())
      {
            int tt=q.front(); q.pop(); inque[tt]=0;
            for (int i=0;i<=t;i++)
                  if ((maxf[tt][i]-flow[tt][i])&&cost[tt]
+value(tt,i)<cost[i])
                  {
                        cost[i]=cost[tt]+value(tt,i);
                        fa[i]=tt;
                        if (!inque[i])
                        {
                              inque[i]=1;
                              q.push(i);
                        }
                  }
      }
      return cost[t]<2000000000;
}

void change(){
      for(int tt=t;tt!=f;tt=fa[tt]){
            ans+=value(fa[tt],tt);
            flow[fa[tt]][tt]++;
            flow[tt][fa[tt]]--;
      }
}

int main(){
      while (cin>>n>>m&&n&&m)   {
            init();
            while (spfamark())
                  change();
            cout<<ans<<endl;
      }
      return 0;
}
```

```
//状态压缩之棋盘放车
long long dp[1<<20];
int line[20];
int pow[21];

inline int getbit(int x)
{
        int ans=0;
        while (x)
        {
                ans++; x-=(x&-x);
        }
        return ans;
}
int main()
{
        for (int i=0;i<=20;i++)
                pow[i]=1<<i;
        int nn;
        scanf("%d",&nn);
        while (nn--)
        {
                int n;
                scanf("%d",&n);
                memset(line,0,sizeof(line));
                for (int i=0;i<n;i++)
                        for (int j=0;j<n;j++)
                        {
                                int t;
                                scanf("%d",&t);
                                if (t)
                                        line[i]+=pow[j];
                        }
                dp[0]=1;
                for (int i=1;i<pow[n];i++)
                {
                        dp[i]=0;
                        int bit=getbit(i);
                        for (int t=i&line[bit-1],j=t&-t;j>0;t-=j,j=t&-t)
                                dp[i]+=dp[i^j];
                }
                printf("%lld\n",dp[pow[n]-1]);
        }
        return 0;
}
```

面积并：From：POI01 火星地图

```
class segment
{
public:
    int l,r,cover,length;
    segment *lc,*rc;
    segment(int L,int R)
    {
        l=L; r=R; cover=0; length=0;
        if (l<r)
        {
            int m=(L+R)>>1;
            lc=new segment(L,m);
            rc=new segment(m+1,R);
        }
    }
    void insert(int L,int R,int delta)
    {
        if (L<=l&&r<=R)
            cover+=delta;
        else
        {
            if (L<=lc->r)
                lc->insert(L,R,delta);
            if (R>=rc->l)
                rc->insert(L,R,delta);
        }
        if (cover)
            length=r-l+1;
        else
            if (l<r)
                length=lc->length+rc->length;
            else
                length=0;
    }
    int count()
    {
            return length;
    }
};
struct line
{
    int x,y1,y2;
    bool operator <(const line &b) const
    {
        return x<b.x;
    }
};
```

```
segment a(0,30000);
line st[10010],ed[10010];
int x[20010];
int n;

int main()
{
    scanf("%d",&n);
    for (int i=0;i<n;i++)
    {
        int x1,y1,x2,y2;
        scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
        st[i].x=x1;st[i].y1=y1;st[i].y2=y2;
        ed[i].x=x2;ed[i].y1=y1;ed[i].y2=y2;
        x[i]=x1;
        x[i+n]=x2;
    }
    sort(x,x+n+n);
    sort(st,st+n);
    sort(ed,ed+n);
    long long ans=0;
    int ST=0,ED=0;
    for (int i=0;i<n+n;i++)
    {
        if (i)
        {
            if (x[i]==x[i-1])
                continue;
            ans+=a.count()*(x[i]-x[i-1]);
        }
        for (;ST<n&&st[ST].x==x[i];ST++)
            a.insert(st[ST].y1,st[ST].y2-1,1);
        for (;ED<n&&ed[ED].x==x[i];ED++)
            a.insert(ed[ED].y1,ed[ED].y2-1,-1);
    }
    printf("%d\n",ans);
    return 0;
}
```

周长并：POJ1177

```
line inX[10010]; line ouX[10010]; line inY[10010]; line ouY[10010];
int n; int ans=0;
void work(line in[],line ou[])
{
     int y=-10000; int i,j;
     for (i=0,j=0;i<n&&j<n;)
     {
         while (y<in[i].y && y<ou[j].y)  y++;
         for (;i<n && in[i].y==y;i++)
         {
               int last=root.length;
               root.insert(in[i].x1,in[i].x2-1,1);
               ans+=abs(root.length-last);
         }
         for (;j<n && ou[j].y==y;j++)
         {
               int last=root.length;
               root.insert(ou[j].x1,ou[j].x2-1,-1);
               ans+=abs(last-root.length);
         }
     }
     for (;j<n;j++)
     {
         int last=root.length;
         root.insert(ou[j].x1,ou[j].x2-1,-1);
         ans+=abs(last-root.length);
     }
}
int main()
{
    scanf("%d",&n);
    for (int i=0;i<n;i++)
    {
        int x1,y1,x2,y2;
        scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
        //x2>x1,y2>y1
        inX[i].x1=x1; inX[i].x2=x2; inX[i].y=y1;
        ouX[i].x1=x1; ouX[i].x2=x2; ouX[i].y=y2;
        inY[i].x1=y1; inY[i].x2=y2; inY[i].y=x1;
        ouY[i].x1=y1; ouY[i].x2=y2; ouY[i].y=x2;
    }
    sort(inX,inX+n); sort(inY,inY+n);
    sort(ouX,ouX+n); sort(ouY,ouY+n);
    work(inX,ouX);
    work(inY,ouY);
    printf("%d\n",ans);
    return 0;
}
```

```
//source：POJ1273 预流推进，n^3
const int inf=2000000000;
int c[210][210];
int f[210][210];
int e[210];
int h[210];
int S,T;
queue<int> q;
bool inque[210];
void init(){
    h[S]=T;
    e[S]=inf;
    for (int i=S;i<=T;i++)
        if (c[S][i])
        {
            f[S][i]=c[S][i];
            f[i][S]=-c[S][i];
            e[i]+=c[S][i];
            e[S]-=c[S][i];
            if (i!=S&&i!=T)
            {
                q.push(i);
                inque[i]=1;
            }
        }
}
void push(int k){
    for (int i=S;i<=T&&e[k]>0;i++) {
        if (c[k][i]-f[k][i]>0&&h[k]==h[i]+1) {
            int delta=min(e[k],c[k][i]-f[k][i]);
            f[k][i]+=delta;
            f[i][k]-=delta;
            e[k]-=delta;
            e[i]+=delta;
            if (!inque[i]&&i!=S&&i!=T&&e[i]>0){
                inque[i]=1;
                q.push(i);
            }
        }
    }
}
void relable(int k)
{
     int tmp=inf;
     for(int i=S;i<=T;i++)
          if(i!=k&&h[i]<tmp&&c[k][i]-f[k][i]>0)
               tmp=h[i];
        h[k]=tmp+1;
}
```

```
int main()
{
    int n,m;
    while (scanf("%d%d",&m,&n)!=EOF)
    {
        memset(c,0,sizeof(c));
        memset(f,0,sizeof(f));
        memset(e,0,sizeof(e));
        memset(h,0,sizeof(h));
        memset(inque,0,sizeof(inque));
        S=1;T=n;
        while (m--)
        {
            int f,t,w;
            scanf("%d%d%d",&f,&t,&w);
            c[f][t]+=w;
        }
        init();
        while (!q.empty())
        {
            int tt=q.front(); q.pop();
            while (e[tt])
            {
                push(tt);
                if (e[tt])
                    relable(tt);
            }
            inque[tt]=0;
        }
        printf("%d\n",e[T]);
    }
    return 0;
}
```

```
#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <cmath>
#include <ctime>
#include <cassert>
const int MAXN = 510;

int N,M;
int adj[MAXN][MAXN];

bool visit[MAXN];
int dist[MAXN];
int vec[MAXN];
int work() {
    if (N == 2) return adj[vec[0]][vec[1]];
    memset(visit,0,sizeof(visit));
    visit[vec[0]] = true; int cnt = 1;
    for (int i = 0; i < N; i++) {
        dist[vec[i]] = adj[vec[0]][vec[i]];
    }
    int p1 = 0; int p2 = 0;
    for (int nn = N - 1; nn ; nn --) {
        int pos = vec[0];
        for (int i = 0; i < N; i++) {
            if (visit[vec[i]]) continue;
            if (dist[vec[i]] > dist[pos]) pos = vec[i];
        }
        p2 = p1; p1 = pos;
        visit[pos] = true; cnt ++;
        for (int i = 0; i < N; i++) {
            if (visit[vec[i]]) continue;
            dist[vec[i]] += adj[pos][vec[i]];
        }
    }
    if (cnt < N) return 0;
    int ans = dist[p1];
    for (int i = 0; i < N; i++) {
        if (adj[vec[i]][p1] && vec[i] != p2) {
            adj[vec[i]][p2] += adj[vec[i]][p1];
            adj[p2][vec[i]] += adj[p1][vec[i]];
        }
    }
    for (int i = 0; i < N; i++) {
        if (vec[i] == p1) {
            vec[i] = vec[--N];
            break;
        }
    }
    int tmp = work();
    return tmp > ans ? ans : tmp;
}
```

```
#include <cstdio>
#include <cstring>
using namespace std;
const unsigned int maxn=128,NOEDGE=~0;
unsigned int G[maxn][maxn];
int N,M;
int res;
template <class T>
void update(T& o,const T& x){
    if(o>x)
        o=x;
}
bool vis[maxn];
void dfs(int v){
    vis[v]=true;
    for(int i=2;i<=N;++i)
        if((!vis[i])&&G[v][i]!=NOEDGE)
            dfs(i);
}
bool possible(){
    memset(vis,0,sizeof(vis));
    dfs(1);
    for(int i=2;i<=N;++i)
        if(!vis[i])
            return false;
    return true;
}
int pre[maxn];
bool del[maxn];
void solve(){
    int num=N;
    memset(del,0,sizeof(del));
    for(;;){
        int i;
        for(i=2;i<=N;++i){
            if(del[i])continue;
            pre[i]=i;
            G[i][i]=NOEDGE;
            for(int j=1;j<=N;++j){
                if(del[j])continue;
                if(G[j][i]<G[pre[i]][i])
                    pre[i]=j;
            }
        }
        for(i=2;i<=N;++i){
            if(del[i])continue;
            int j=i;
            memset(vis,0,sizeof(vis));
            while(!vis[j]&&j!=1){
                vis[j]=true;
                j=pre[j];
            }
            if(j==1)continue;
            i=j;
            res+=G[pre[i]][i];
            for(j=pre[i];j!=i;j=pre[j]){
                res+=G[pre[j]][j];
                del[j]=true;
            }
            for(j=1;j<=N;++j){
                if(del[j])continue;
                if(G[j][i]!=NOEDGE)
                    G[j][i]-=G[pre[i]][i];
            }
            for(j=pre[i];j!=i;j=pre[j]){
                for(int k=1;k<=N;++k){
                    if(del[k])continue;
                    update(G[i][k],G[j][k]);
                    if(G[k][j]!=NOEDGE)
                        update(G[k][i],G[k][j]-G[pre[j]][j]);
                }
            }
            for(j=pre[i];j!=i;j=pre[j]){
                del[j]=true;
            }
            break;
        }
        if(i>N){
            for(int i=2;i<=N;++i){
                if(del[i])continue;
                res+=G[pre[i]][i];
            }
            break;
        }
    }
}
```