ACM-ICPC

4

1.0.0

哈尔滨理工大学 ACM-ICPC 集训队

2012 年 12 月

2012 5 ACM-ICPC

2011 10

OJ OJ

2011 9 ACM-ICPC

2011-2012

2011

——

2011 9

OJ

2012 8 2012 ACM-ICPC

ACM-ICPC

BBS

ACM-ICPC

2012 10

ACM-ICPC

OJ

acm@hrbust.edu.cn

Hrbust-OJ http://acm.hrbust.edu.cn

AC

OJ

http://acm.hdu.edu.cn

http://poj.org

http://acm.zju.edu.cn

OJ:

http://hi.baidu.com/leo_xxx/item/6719a5ffe25755713c198b50

ACM-ICPC

2012 12

目 录

第3章 图论

3.1 图的基本概念

V

n*(n-1)/2

n-1

0

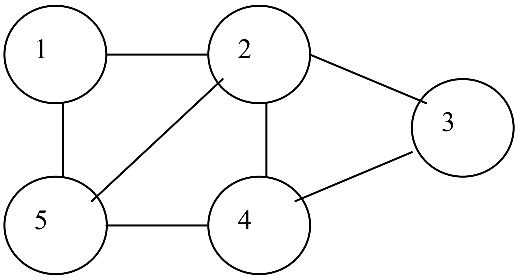
3.1.1 图的表示

graph[N][N] graph[i][j] i

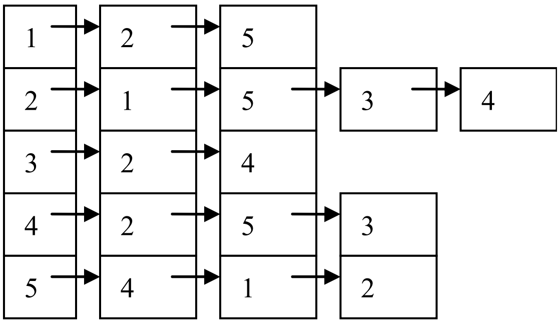
j 0 graph[i][j]

graph[i][j]==graph[j][i]

bool char



1-1



1-2

O(n^2) O(V+E)

```
struct Arc
{
   int next_arc;
   int point;
};
int node[V];
struct Arc arc[E];
```

node arc node[i] i

arc next_arc arc

node[i]

```
void AddEdge(int u,int v)
{
   arc[EdgeCount].next_arc=node[u];
   arc[EdgeCount].point=v;
   node[u]=EdgeCount;
   EdgeCount++;
}
```

EdgeCount 0.

3.1.2 广度优先搜索

1 1

1 2 5 2 3 4 5 5 4

1 2 5 1 2 2

3 4 “ ”

O(V+E)

1.1.1

```
int que[V];//
int vis[V];//        vis[i]==1   i       0
int front,rear;//

void bfs()
{
   front=rear=0;
   memset(vis,0,sizeof(vis));

   que[rear++]=0;//     0                            0
   //
```

```
    vis[0]=1;

    while(front<que)
    {
        int cur_node=que[front++];//
        int edge;
        //
        for(edge=node[cur_node];edge!=-1;edge=arc[edge].next_arc)
        {
            if(!vis[arc[edge].point])
            {
                que[rear++]=arc[edge].point;
                vis[arc[edge].point]=1;
            }
        }
    }
}
```

3.1.3 深度优先搜索

1 2 3 4 5
4 5

O(V+E)

```
    int vis[V];
    void dfs(int v)
    {
        vis[v]=1;
        int edge;
        for(edge=node[v];edge!=-1,edge=arc[edge].next_arc)
        {
            if(!vis[arc[edge].point])
            {
                dfs(arc[edge].point);
            }
        }
    }
```

3.2 拓扑排序

3.2.1 基本原理

(Direct Acyclic Graph, DAG)
(u,v) u
v

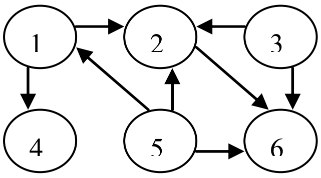
DAG

0
0

0

1 0

DAG u,v
v u DFS u,v u v
DFS DFS



1.2.1

1.2.1

0 3 5
3 2 6 1
5 1 2 6 1 1 0
1 2 4 1 2 4 0.
2 6 1 6 0
4 4
6 6
3 5 1 2 4 6.

1
1 2 6 4 1 4 2 6

3 5 5 3
3 5 1 2 6 4

3.2.2 模板代码

```
struct Arc
{
    int point;
    int next_arc;
};

Arc arc[50005];
int node[5005];
int digree[5005];
int top[5005];

int main()
{
 int n,m;
 scanf("%d%d",&n,&m);
    queue<int>q;
```

```
   for(int i=1;i<=m;i++)
   {
      int a,b;
      scanf("%d%d",&a,&b);

      arc[i].next=node[a];
      arc[i].point=b;
      node[a]=i;
      digree[b]++;
   }

   for(int i=1;i<=n;i++)
   {
      if(digree[i]==0)
      {
         q.push(i);
      }
   }
   int l=0;
   while(!q.empty())
   {
      int x=q.front();
      top[l++]=x;//  x
      q.pop();
      for(int e=node[x];e!=-1;e=arc[e].next)
      {
         digree[arc[e].point]--;
         if(digree[arc[e].point]==0)
         {
            q.push(arc[e].point);
         }
      }
   }
 return 0;
}
```

3.2.3 经典题目

1 /

HDOJ 1285

2

N 1<=N<=500 1 2 3 N

P1 P2 P1 P2 P1 P2

3

4

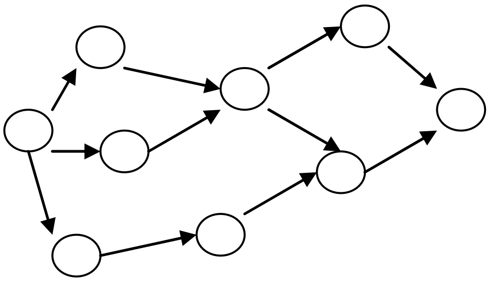
```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<queue>
using namespace std;

int graph[505][505];//
int digree[505];//

int main()
{
   int n,m;
   while(scanf("%d%d",&n,&m)!=EOF)
   {
      memset(graph,0,sizeof(graph));
      memset(digree,0,sizeof(digree));

      for(int i=0;i<m;i++)
      {
         int u,v;
         scanf("%d%d",&u,&v);
         if(!graph[u][v])//
         {
            graph[u][v]=1;
            digree[v]++;
         }
      }

      priority_queue<int,vector<int>,greater<int> >q;//    STL
                                                               >
      for(int i=1;i<=n;i++)
      {
         if(digree[i]==0)q.push(i);
      }

      bool first=1;
      while(!q.empty())
      {
         int cur=q.top();
         q.pop();
         if(first){cout<<cur;first=0;}
         else cout<<' '<<cur;

         for(int i=1;i<=n;i++)
         {
            if(graph[cur][i])
            {
               digree[i]--;
               if(digree[i]==0)
               {
                  q.push(i);
               }
            }
         }
      }
      printf("\n");
   }
   return 0;
}
```

3.3 活动网络(AOE 网络)

AOE

AOE

DAG



a[i] i a[0] 0
0

+ 0
i (i,j) j 1
a[j]=MAX(a[j],a[i]+g[i][j]);g[i][j] (i,j)

3.4 最小生成树 Prim

V V-1

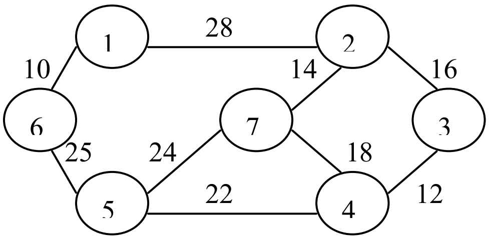
N-1

3.4.1 基本原理

Prim V
S V-S S S
V-S S
N

graph[N][N] low[N]

low



1.4.1

1

6 6

6 5 25 5
4 3 2 7

N
Prim O(N^2)
Prim
Prim

3.4.2 模板代码

```
#define INF 0x1f1f1f1f//
//prim                         n-1
//   cost                 n
int prim(int cost[][200],int n)
{
   //low                              vis
   int low[10000],vis[10000]={0};
   int i,j,p;
   int min,res=0;
   vis[0]=1;
   for(i=1;i<n;i++)low[i]=cost[0][i];
   for(i=1;i<n;i++)
   {
      min=INF;p=-1;
      for(j=0;j<n;j++)
      {
         if(0==vis[j]&&min>low[j])
         {
            min=low[j];
            p=j;
         }
      }
      //min==INF
      if(min==INF)return -1;

      res+=min;
      vis[p]=1;

      for(j=0;j<n;j++)
      {
         if(0==vis[j]&&low[j]>cost[p][j])
         {
            low[j]=cost[p][j];
         }
```

```
            }
        }
        return res;
    }
```

3.4.3 经典题目

3.4.3.1 题目 1

1 /

HDOJ 1102

2

N 1-N

A B C C A C B

A B

3

0 prim

4

```
#include<iostream>
#include<cstdio>
using namespace std;
#define INF 0x1f1f1f1f//
//prim                          n-1
//   cost              n
int prim(int cost[][200],int n)
{
    //low                          vis
    int low[10000],vis[10000]={0};
    int i,j,p;
    int min,res=0;
    vis[0]=1;
    for(i=1;i<n;i++)low[i]=cost[0][i];
    for(i=1;i<n;i++)
    {
        min=INF;p=-1;
        for(j=0;j<n;j++)
        {
            if(0==vis[j]&&min>low[j])
            {
                min=low[j];
                p=j;
            }
        }
        //min==INF
        if(min==INF)return -1;

        res+=min;
        vis[p]=1;

        for(j=0;j<n;j++)
        {
            if(0==vis[j]&&low[j]>cost[p][j])
            {
                low[j]=cost[p][j];
            }
        }
    }
    return res;
}

int main()
{
    int n;
    int a[200][200];
```

```
        while(cin>>n)
        {
            for(int i=0;i<n;i++)
            {
                for(int j=0;j<n;j++)
                {
                    scanf("%d",&a[i][j]);
                }
            }
            int q;
            scanf("%d",&q);
            for(int i=0;i<q;i++)
            {
                int u,v;
                scanf("%d%d",&u,&v);
                a[u-1][v-1]=a[v-1][u-1]=0;
            }
            int res=prim(a,n);
            printf("%d\n",res);
        }
        return 0;
    }
```

3.4.3.2 题目 2

1 /

HDOJ 2489

2

Ratio

$$Ratio = \frac{\sum edge\ weight}{\sum node\ weight}$$

n m

Ratio m

3

m Ratio

n m

m Ratio m Ratio

4

```
#include<set>
#include<map>
#include<cmath>
#include<queue>
#include<stack>
#include<vector>
#include<cstdio>
#include<cstring>
#include<iostream>
#include<algorithm>

using namespace std;

#define INF 0x1f1f1f1f
#define MIN(a,b) ((a)<(b)?(a):(b))
#define MAX(a,b) ((a)>(b)?(a):(b))

int edge_wei[20][20];//
int node_wei[20];//
bool flag[20];
bool fl[20];
```

```
double res;
int dfs_cnt;
bool arr[20];//result array

int n,m;

int mst()//
{
 int ret=0;
 int low[20]={0};
 int sta;
 for(int i=1;i<=n;i++)
 {
     if(fl[i])
     {
         sta=i;
         break;
     }
 }
 low[sta]=0;
 flag[sta]=1;
 for(int i=1;i<=n;i++)
 {
     if(fl[i])
     {
         low[i]=edge_wei[sta][i];
     }
 }
 for(int i=1;i<m;i++)
 {
     int min=INF;
     int loc;
     for(int j=1;j<=n;j++)
     {
         if(fl[j]&&!flag[j]&&low[j]<min)
         {
             min=low[j];
             loc=j;
         }
     }
     flag[loc]=1;
     ret+=low[loc];
     for(int j=1;j<=n;j++)
     {
         if(fl[j]&&!flag[j])
         {
             if(edge_wei[loc][j]<low[j])
             {
                 low[j]=edge_wei[loc][j];
             }
         }
     }
 }
 return ret;
}

void dfs(int v)
{
 fl[v]=1;
 dfs_cnt++;
 if(dfs_cnt==m)
 {
     memset(flag,0,sizeof(flag));
     int r=mst();
     int sum=0;
     for(int i=1;i<=n;i++)
     {
         if(fl[i])
         {
             sum+=node_wei[i];
         }
     }
     double res_tmp=(double)r/(double)sum;
     if(res_tmp-res<-(1e-9))//
     {
         res=res_tmp;
         for(int i=1;i<=n;i++)
         {
             arr[i]=fl[i];
         }
     }
     fl[v]=0;
     dfs_cnt--;
     return;
 }
 for(int i=v+1;i<=n;i++)
 {
     dfs(i);
 }
 fl[v]=0;
 dfs_cnt--;
}

int main()
{
 while(scanf("%d%d",&n,&m),n||m)
 {
     for(int i=1;i<=n;i++)
     {
         scanf("%d",&node_wei[i]);
     }
     for(int i=1;i<=n;i++)
     {
         for(int j=1;j<=n;j++)
         {
             scanf("%d",&edge_wei[i][j]);
         }
     }

     res=(double)INF;
     for(int i=1;i<=n-m+1;i++)
     {
         memset(fl,0,sizeof(fl));
         dfs_cnt=0;
         dfs(i);
     }

     int fir=1;
     for(int i=1;i<=n;i++)
     {
         if(arr[i])
         {
             if(fir)
             {
                 fir=0;
                 printf("%d",i);
             }
             else
             {
                 printf(" %d",i);
             }
         }
     }
     printf("\n");
 }
 return 0;
}
```

3.5 最小生成树 Kruskal

3.5.1 基本原理

Kruskal

V-1

kruskal

O(ElogV)

O(E) prim O(N^2)

Prim Kruskal

3.5.2 模板代码

```
int p[10005];//
int r[10005];//

//
int find(int v)
{
   if(v!=p[v])p[v]=find(p[v]);
   return p[v];
}

//
void join(int u,int v)
{
   int a=find(u);
   int b=find(v);
   if(a==b)return;

   if(r[a]<r[b])
   {
      p[a]=b;
   }
   else if(r[a]>r[b])
   {
      p[b]=a;
   }
   else
   {
      p[a]=b;
      r[b]++;
   }
}

//
void init_set(int n)
{
   int i;
   for(i=1;i<=n;i++)
   {
      p[i]=i;
      r[i]=1;
   }
}
```

```
//
struct Edge
{
   int u;
   int v;
   int weight;
};

struct Edge edge[50005];

//                 qsort   sort
void quick_sort(struct Edge* start,struct Edge* end)
{
   if(start>=end)return;
   struct Edge* loc=start;
   struct Edge* iterator;
   struct Edge tmp;
   for(iterator=start;iterator!=end;iterator++)
   {
      if(iterator->weight<(end-1)->weight)
      {
         tmp=*loc;
         *loc=*iterator;
         *iterator=tmp;

         loc++;
      }
   }
   tmp=*loc;
   *loc=*(end-1);
   *(end-1)=tmp;

   quick_sort(start,loc);
   quick_sort(loc+1,end);
}

int kru(int n,int m)//kruskal                n        m
{
   init_set(n);
   quick_sort(edge,edge+m);
   int i;
   int ret=0;//
   int cnt=0;//
   for(i=0;i<m;i++)
   {
      int u=edge[i].u;
      int v=edge[i].v;
      //
      if(find(u)!=find(v))
      {
         cnt++;
         ret+=edge[i].weight;
         join(u,v);
      }
      if(cnt==n-1)return ret;//     n-1
   }
   return -1;
}
```

3.5.3 经典题目

1 /

POJ 1797

2

1 n

3

1 n

4

kruskal

```
#include<iostream>
#include<cstdio>
#include<algorithm>
using namespace std;

struct Edge
{
   int u;
   int v;
   int w;
};

bool cmp(Edge e1,Edge e2)
{
   return e1.w>e2.w;
}

Edge e[1000005];

int p[1005];
int r[1005];

int find(int u)
{
   if(u!=p[u])p[u]=find(p[u]);
   return p[u];
}

void join(int u,int v)
{
   int a=find(u);
   int b=find(v);
   if(a==b)return;
   if(r[a]>r[b])p[b]=a;
   else if(r[a]<r[b])p[a]=b;
   else
   {
      p[a]=b;
      r[b]++;
   }
}

int main()
{
   int t;
   int cse=1;
   scanf("%d",&t);
   while(t--)
   {
      int n,m;
      scanf("%d%d",&n,&m);
      for(int i=1;i<=n;i++)
      {
         p[i]=i;
         r[i]=1;
      }
      for(int i=0;i<m;i++)
      {
         scanf("%d%d%d",&e[i].u,&e[i].v,&e[i].w);
```

```
        }
        sort(e,e+m,cmp);
        int res;
        for(int i=0;i<m;i++)
        {
            if(find(1)==find(n))break;
            int u=e[i].u;
            int v=e[i].v;
            if(find(u)!=find(v))
            {
                res=e[i].w;
                join(u,v);
            }
        }
        printf("Scenario #%d:\n%d\n\n",cse++,res);
    }
    return 0;
}

#include<iostream>
#include<queue>
#include<cstdio>
#include<cstring>
using namespace std;

int n,m;
int edge_cnt;

int head[1005];
int next[1000005];
int point[1000005];
int weight[1000005];

int dis[1005];//dis    1
bool fl[1005];

void add_edge(int u,int v,int w)
{
    next[edge_cnt]=head[u];
    point[edge_cnt]=v;
    weight[edge_cnt]=w;
    head[u]=edge_cnt;
}

void fun()
{
    memset(dis,0,sizeof(dis));
    memset(fl,0,sizeof(fl));

    queue<int>q;

    q.push(1);
    fl[1]=1;
    dis[1]=0x1f1f1f1f;//                             0

    while(!q.empty())
    {
        int u=q.front();
        q.pop();
        fl[u]=0;

        for(int e=head[u];e!=-1;e=next[e])
        {
            int v=point[e];
            if(weight[e]<dis[u])//          u     dis          u     v

            {
                if(weight[e]>dis[v])//                 v     dis             v   dis
weight[e]      v   dis
                {
                    dis[v]=weight[e];
                    if(!fl[v])
                    {
                        q.push(v);
                    }
                }
            }
            else if(dis[v]<dis[u])//              v   dis      v   dis     dis[u]  dis[v]

            {
                dis[v]=dis[u];
                if(!fl[v])
                {
                    q.push(v);
                }
            }
        }
    }
}

int main()
{
    int t;
    scanf("%d",&t);
    int sen=1;
    while(t--)
    {
        memset(head,-1,sizeof(head));
        edge_cnt=0;
        scanf("%d%d",&n,&m);
        for(int i=1;i<=m;i++)
        {
            int u,v,w;
            scanf("%d%d%d",&u,&v,&w);
            edge_cnt++;
            add_edge(u,v,w);
            edge_cnt++;
            add_edge(v,u,w);
        }
        fun();
        printf("Scenario #%d:\n%d\n\n",sen++,dis[n]);
    }
    return 0;
}
```

3.6 最短路 Dijkstra

3.6.1 基本原理

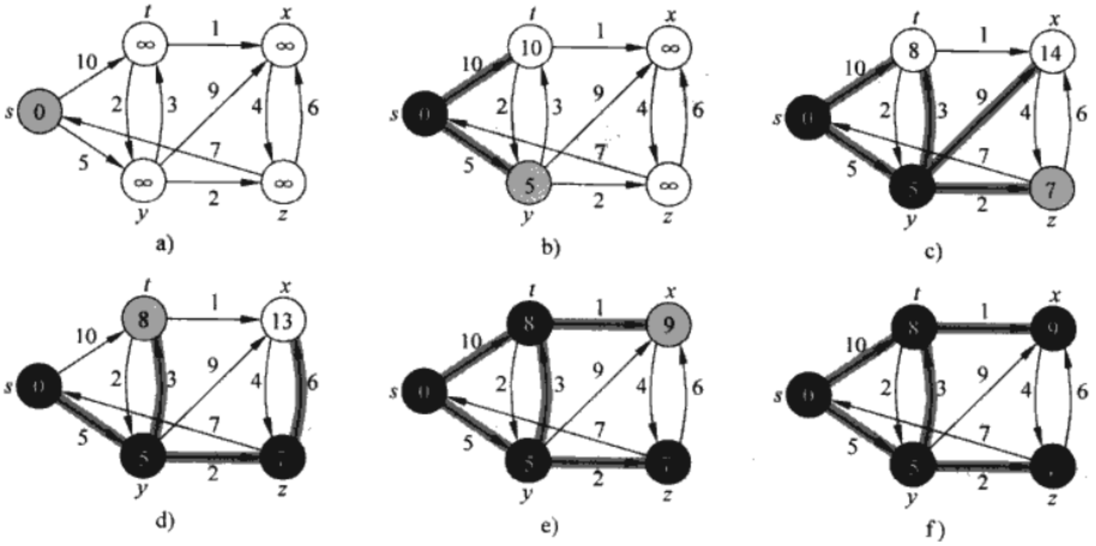
v0, v0 v0 v0
v0 v0
v0 v0
v1, v0 v1 (v0,v1)
v0 v0 v1
v0 v1

v1 v0 v0
v2, v2 v0
v0 v2 v0

1.
2.
n-1 n-1

Dijkstra
1. 0

2.
v1 v2 (v1,v2) +
v2



Dijkstra
s s y 5 t 10 y
y y s +(y,t) t s
t 8 z 7 x 14
s z 7 s
Dijkstra

3.6.2 模板代码

```
#define INF 0xfffffff
#define SIZE 150

int a[SIZE][SIZE]
int low[SIZE];

void DIJ(int n)//          n      0
{
    int i,j,k;
    low[0]=0;
    bool flag[SIZE]={0};
    flag[0]=1;
    for(i=1;i<n;i++)
    {
        low[i]=a[0][i];
    }

    for(i=1;i<n;i++)
    {
        int min=INF;
        for(j=0;j<n;j++)
        {
            if(flag[j]==0&&low[j]<min)
            {
                min=low[j];
                k=j;
            }
        }
        flag[k]=1;
        for(j=0;j<n;j++)
        {
            if(flag[j]==0&&a[k][j]+low[k]<low[j])
                low[j]=low[k]+a[k][j];
        }
    }
}

struct Arc
{
    int next_arc;
    int point;
    int weight;
};
int node[N];
struct Arc arc[M];

void insert_edge(int u,int v,int weight,int edge_num)
{
    arc[edge_num].next_arc=node[u];
    arc[edge_num].point=v;
    arc[edge_num].weight=weight;
    node[u]=edge_num;
}
//
struct heap_elm
{
    int num;
    int dis;
};
//
struct heap_elm heap[M];
//                                       len
void insert(struct heap_elm h,int* len)
{
    (*len)++;
    heap[*len]=h;
    int i=*len;
    while(i>0)
    {
        int j=(i>>1);
        if(heap[j].dis>heap[i].dis)
        {
            struct heap_elm tmp=heap[j];
```

```
         heap[j]=heap[i];
         heap[i]=tmp;

         i=j;
      }
      else break;
   }
}
//
void heapfi(int loc,int len)
{
   int left=(loc<<1);
   int right=left+1;
   int min_loc=loc;
   if(left<=len&&heap[left].dis<heap[min_loc].dis)
   {
      min_loc=left;
   }
   if(right<=len&&heap[right].dis<heap[min_loc].dis)
   {
      min_loc=right;
   }
   if(min_loc!=loc)
   {
      struct heap_elm tmp=heap[min_loc];
      heap[min_loc]=heap[loc];
      heap[loc]=tmp;

      heapfi(min_loc,len);
   }
}
//
struct heap_elm get()
{
   return heap[1];
}
//
void del(int* len)
{
   heap[1]=heap[*len];
   (*len)--;
   heapfi(1,*len);
}
//
int vis[N];
//             res[i]        i
int res[N];
//Dijkstra                     n     m      src
void dij(int n,int m,int src)
{
   memset(vis,0,sizeof(vis));
   int len=0;//
   struct heap_elm h;
   h.dis=0;
   h.num=src;
   insert(h,&len);//
   while(len>0)
   {
      h=get();
      del(&len);
      if(vis[h.num])continue;//
      res[h.num]=h.dis;
      vis[h.num]=1;
      int edge;
      for(edge=node[h.num];edge!=-1;edge=arc[edge].next_arc)
      {
         if(!vis[arc[edge].point])
         {
            struct heap_elm t;
            t.dis=h.dis+arc[edge].weight;
            t.num=arc[edge].point;
            insert(t,&len);
         }
      }
   }
}
```

3.6.3 经典题目

3.6.3.1 题目 1

1 /

HDOJ 3790

2

d p s t

3

Dijkstra

+ SPFA

4

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

#define INF 0x1f1f1f1f
#define SIZE 1005

int dis[SIZE][SIZE];
int cost[SIZE][SIZE];
int lowdis[SIZE];//
int lowcost[SIZE];//

void DIJ(int n,int s,int t)
{
   bool flag[SIZE]={0};
   flag[s]=1;
   for(int i=1;i<=n;i++)
   {
      lowdis[i]=dis[s][i];
      lowcost[i]=cost[s][i];
   }
   int k;
   for(int i=1;i<n;i++)
   {
      int mindis=INF;
      int mincost=INF;
      for(int j=1;j<=n;j++)
      {
         if(flag[j]==0)
         {
            if(lowdis[j]<mindis)
            {
               mindis=lowdis[j];
               mincost=lowcost[j];
               k=j;
            }
            else if(lowdis[j]==mindis&&lowcost[j]<mincost)
            {
               mindis=lowdis[j];
               mincost=lowcost[j];
```

```
                  k=j;
               }
            }
         }
         flag[k]=1;
         for(int j=1;j<=n;j++)
         {
            if(flag[j]==0)
            {
               if(dis[k][j]+lowdis[k]<lowdis[j])
               {
                  lowdis[j]=lowdis[k]+dis[k][j];
                  lowcost[j]=lowcost[k]+cost[k][j];
               }
               else
if(dis[k][j]+lowdis[k]==lowdis[j]&&cost[k][j]+lowcost[k]<lowcost[j])
               {
                  lowdis[j]=lowdis[k]+dis[k][j];
                  lowcost[j]=lowcost[k]+cost[k][j];
               }
            }
         }
      }
   }

   int main()
   {
      int n,m;
      while(scanf("%d%d",&n,&m),n||m)
      {
         memset(dis,0x1f,sizeof(dis));
         memset(cost,0x1f,sizeof(cost));
         for(int i=1;i<=n;i++)
         {
            dis[i][i]=0;
            cost[i][i]=0;
         }
         for(int i=0;i<m;i++)
         {
            int u,v,d,c;
            scanf("%d%d%d%d",&u,&v,&d,&c);
            if(dis[u][v]>d||(dis[u][v]==d&&cost[u][v]>c))
            {
               dis[u][v]=dis[v][u]=d;
               cost[u][v]=cost[v][u]=c;
            }
         }
         int s,t;
         scanf("%d%d",&s,&t);
         DIJ(n,s,t);
         printf("%d %d\n",lowdis[t],lowcost[t]);
      }
      return 0;
   }
```

3.6.3.2 题目 2

1 /

hrbust 1339

2

3

Y Y

4

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

#define INF 0x1f1f1f1f//

struct str//
{
 int num;//
 int cost;//
 str(int n,int c):num(n),cost(c){}//    C++
 str(){}
 friend bool operator < (str s1,str s2)//
 {
    return s1.cost>s2.cost;
 }
};

struct Arc
{
   int next_arc;
   int point;
   int cost;
};

Arc arc[20005];
int head[5005];
bool fl[5005];

int lowa[5005];
int lowb[5005];
int lowc[5005];
int C,A,B;

void dij(int src,int n,int* low)//low
{
   memset(fl,0,sizeof(fl));//
   priority_queue<str>q;//       STL
   q.push(str(src,0));
   int kk=0;
   while(kk<n&&!q.empty())
   {
      str s=q.top();
      q.pop();
      if(fl[s.num])continue;
      fl[s.num]=1;
      low[s.num]=s.cost;
      kk++;
      for(int e=head[s.num];e!=-1;e=arc[e].next_arc)
      {
         if(!fl[arc[e].point])
         {
            q.push(str(arc[e].point,arc[e].cost+s.cost));
         }
      }
   }
}
```

```
int main()
{
 int cse=1;
 int n,m;
 while(~scanf("%d%d",&n,&m))
 {
    memset(head,-1,sizeof(head));

    memset(lowc,0x1f,sizeof(lowc));
    memset(lowa,0x1f,sizeof(lowa));
    memset(lowb,0x1f,sizeof(lowb));

    scanf("%d%d%d",&C,&A,&B);

    for(int i=1;i<=m;i++)
    {
        int x,y,k;
        scanf("%d%d%d",&x,&y,&k);

        /////
        arc[i].next_arc=head[x];
        arc[i].point=y;
        arc[i].cost=k;
        head[x]=i;

         /////
        arc[m+i].next_arc=head[y];
        arc[m+i].point=x;
        arc[m+i].cost=k;
        head[y]=m+i;
    }

    dij(C,n,lowc);
    dij(A,n,lowa);
    dij(B,n,lowb);


    printf("Scenario #%d\n",cse++);
    int res=INF;
    if(lowc[B]>=INF||lowc[A]>=INF)
    {
        printf("Can not reah!\n");
        continue;
    }
    for(int i=1;i<=n;i++)
    {
        if(lowc[i]+lowa[i]+lowb[i]<res)
        {
            res=lowc[i]+lowa[i]+lowb[i];
        }
    }
    printf("%d\n",res);
    printf("\n");
 }
 return 0;
}
```

3.7 最短路 Bellman-Ford

3.7.1 基本原理

v0 Bellman-Ford N-1
(u,v) v0 u (u,v) v0 v

v0 v Dijkstra

Bellman-Ford **N-1** **E**
O(NE) N-2
Bellman-Ford N-1

Bellman-Ford
SPFA
N N

Bellman-Ford

3.7.2 模板代码

```
Bellman-Ford

#define N 105

int res[N];//
int g[N][N];

void bellman(int n,int src)
{
   //
   memset(res,0x1f,sizeof(res));
   res[src]=0;//             0
   int i,j,k;
   for(i=1;i<n;i++)//    n-1
   {
      //j  k
      for(j=1;j<=n;j++)
      {
         for(k=1;k<=n;k++)
         {
            if(res[k]>res[j]+g[j][k])
            {
               res[k]=res[j]+g[j][k];
            }
         }
      }
   }
}

int res[N];//
struct Edge
{
   int u;
   int v;
   int t;
};

Edge edge[E];

bool bellman(int n,int m,int src)//n      m      src
{
   //
```

```
    memset(res,0x1f,sizeof(res));
    res[src]=0;//             0
    for(int i=0;i<n;i++)
    {
       for(int j=0;j<m;j++)
       {
          if(res[edge[j].u]+edge[j].t<res[edge[j].v])
          {
             res[edge[j].v]=res[edge[j].u]+edge[j].t;
          }
       }
    }
    for(int i=0;i<n;i++)//
    {
       for(int j=0;j<m;j++)
       {
          if(res[edge[j].u]+edge[j].t<res[edge[j].v])
          {
             return 1;
          }
       }
    }
    return 0;
}
SPFA

#define N 105

int res[N];//
int g[N][N];
int cnt[N];//

int que[N*N];//
bool in_que[N];//
int front;//
int rear;//

void spfa(int n,int src)
{
    rear=front=0;
    que[++rear]=src;
    memset(res,0x1f,sizeof(res));
    memset(in_que,0,sizeof(in_que));
    res[src]=0;
    while(front<rear)
    {
       int cur=que[++front];
       in_que[cur]=0;
       int i;
       for(i=1;i<=n;i++)
       {
          if(res[cur]+g[cur][i]<res[i])
          {
             res[i]=res[cur]+g[cur][i];
             if(!in_que[i])
             {
                que[++rear]=i;
             }
          }
       }
    }
}

int res[505];//
struct Arc
{
    int next_arc;
    int point;
    int t;
};
int node[5000];
struct Arc arc[6000];
int cnt[505];//
bool fl[505];//

bool spfa(int n,int src)//              bool
{
    //
    memset(res,0x1f,sizeof(res));
    memset(cnt,0,sizeof(cnt));
    memset(fl,0,sizeof(fl));

    res[src]=0;//              0
    queue<int>q;//      STL
    q.push(src);//
    cnt[src]++;//          +1
    fl[src]=1;//
    while(!q.empty())
    {
        int c=q.front();
        q.pop();
        fl[c]=0;//
        for(int e=node[c];e!=-1;e=arc[e].next_arc)
        {
            if(res[c]+arc[e].t<res[arc[e].point])
            {
                res[arc[e].point]=res[c]+arc[e].t;
                if(!fl[arc[e].point])
                {
                    q.push(arc[e].point);
                    cnt[arc[e].point]++;
                    if(cnt[arc[e].point]>=n)return 1;
                }
            }
        }
    }
    return 0;
}
```

3.7.3 经典题目

1 /

poj 3259

2

N M W

3

Bellman-Ford SPFA

4

Bellman-Ford:

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

int res[505];//
struct Edge
{
    int u;
    int v;
    int t;
};

Edge edge[6000];

bool bellman(int n,int m,int src)//              bool
{
    //
    memset(res,0x1f,sizeof(res));
    res[src]=0;//              0
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<m;j++)
        {
            if(res[edge[j].u]+edge[j].t<res[edge[j].v])
            {
                res[edge[j].v]=res[edge[j].u]+edge[j].t;
            }
        }
    }
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<m;j++)
        {
            if(res[edge[j].u]+edge[j].t<res[edge[j].v])
            {
                return 1;
            }
        }
    }
    return 0;
}

int main()
{
    int f;
    scanf("%d",&f);
    while(f--)
    {
        int n,m,w;
        scanf("%d%d%d",&n,&m,&w);
        for(int i=0;i<m;i++)
        {
            int s,e,t;
            scanf("%d%d%d",&s,&e,&t);
            edge[i].u=s,edge[i].v=e,edge[i].t=t;
            edge[i+m].u=e,edge[i+m].v=s,edge[i+m].t=t;
        }
        for(int i=0;i<w;i++)
        {
            int s,e,t;
            scanf("%d%d%d",&s,&e,&t);
            edge[i+2*m].u=s,edge[i+2*m].v=e,edge[i+2*m].t=-t;
        }
        if(bellman(n,2*m+w,1))printf("YES\n");
        else printf("NO\n");
    }
    return 0;
}
```

SPFA:

SPFA

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

#define N 505
```

```
int res[N];//
int g[N][N];
int cnt[N];//

int que[N*N];//
bool in_que[N];//
int front;//
int rear;//

bool spfa(int n,int src)
{
   memset(cnt,0,sizeof(cnt));
   rear=front=0;
   que[++rear]=src;
   cnt[src]++;
   memset(res,0x1f,sizeof(res));
   memset(in_que,0,sizeof(in_que));
   res[src]=0;
   while(front<rear)
   {
      int cur=que[++front];
      in_que[cur]=0;
      int i;
      for(i=1;i<=n;i++)
      {
         if(res[cur]+g[cur][i]<res[i])
         {
            res[i]=res[cur]+g[cur][i];
            if(!in_que[i])
            {
               que[++rear]=i;
               cnt[i]++;
               if(cnt[i]>=n)return 1;
            }
         }
      }
   }
   return 0;
}

int main()
{
   int f;
   scanf("%d",&f);
   while(f--)
   {
      int n,m,w;
      scanf("%d%d%d",&n,&m,&w);
      memset(g,0x1f,sizeof(g));
      for(int i=0;i<m;i++)
      {
         int s,e,t;
         scanf("%d%d%d",&s,&e,&t);
         g[s][e]=g[e][s]=(g[s][e]>t?t:g[s][e]);
      }
      for(int i=0;i<w;i++)
      {
         int s,e,t;
         scanf("%d%d%d",&s,&e,&t);
         g[s][e]=(g[s][e]<-t?g[s][e]:-t);
      }
      if(spfa(n,1))printf("YES\n");
      else printf("NO\n");
   }
   return 0;
}
```

3.8 所有顶点之间的最短路 Floyd

3.8.1 基本原理

n 1 2 … n k i j p i
j k
k p p 1 k-1

k p i~k k~j
k-1
i~j k g[k][i][j]
i~j k g[k][i][j]=min(g[k-1][i][j],g[k-1][i][k]+g[k-1][k][j]) k k-1

g[i][j]=min(g[i][j],g[i][k]+g[k][j])
if(g[i][k]&&g[k][j])g[i][j]=1;

A B B C
A C

3.8.2 模板代码

```
int i,j,k;
for(k=0;k<n;k++)
{
   for(i=0;i<n;i++)
   {
      for(j=0;j<n;j++)
      {
         g[i][j]=min(g[i][j],g[i][k]+g[k][j]);//g
      }
   }
}
```

3.8.3 经典题目

3.8.3.1 题目 1

1 /
acm.hnu.cn/online 12519

2
TOM N N
TOM N

3

＋ ＝
N floyd

4

```
#include<iostream>
#include<cstdio>
```

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

#define INF 0x1f1f1f1f

int n;

int g[205][205];//

int a[205];//

void floyd()
{
   int i,j,k;
   for (k=0;k<n;k++)
      for (i=0;i<n;i++)
         for (j=0;j<n;j++)
            if (g[i][k]+g[k][j]<g[i][j])
               g[i][j]=g[i][k]+g[k][j];
}

int main()
{
   int t;
   scanf("%d",&t);
   while(t--)
   {
      memset(g,0,sizeof(g));
      scanf("%d",&n);
      for(int i=0;i<n;i++)
      {
         scanf("%d",&a[i]);
      }
      for(int i=0;i<n;i++)
      {
         for(int j=0;j<n;j++)
         {
            scanf("%d",&g[i][j]);
            if(g[i][j]==-1)//     -1
            {
               g[i][j]=INF;
            }
         }
      }

      floyd();

      int res=0;//    res

      for(int i=0;i<n-1;i++)
      {
         if(g[a[i]][a[i+1]]>=INF)
         {
            res=INF;
            break;
         }
         res+=g[a[i]][a[i+1]];
      }
      res+=g[a[n-1]][a[0]];
      if(res>=INF)
      {
         printf("impossible\n");
      }
      else
      {
         printf("%d\n",res);
      }
   }
   return 0;
```

}

3.8.3.2 题目 2

1 /

hrbust 1348

2

G

1

2

N M Q N

N<=300

M M<=100000;Q Q<=100000

0,1,2,...,N-1

M x,y,c, x y

c,c>0

Q 0 x x 1 x y x y

N=M=Q=0

"Case #:" # 1

0 x x "ERROR! At point x".

1 x y x y "ERROR! At path x to y"

x

y, "No such path"

3

floyd

0

4

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

int g[305][305];
bool fl[305];//

int n,m,q;

int main()
{
 int cse=1;//
 while(scanf("%d%d%d",&n,&m,&q))
 {
     if(n==0&&m==0&&q==0)break;

     memset(g,0x1f,sizeof(g));
```

```
    memset(fl,0,sizeof(fl));

    printf("Case %d:\n",cse++);
    for(int i=0;i<n;i++)//                 0
    {
        g[i][i]=0;
    }
    for(int i=0;i<m;i++)
    {
        int a,b,w;
        scanf("%d%d%d",&a,&b,&w);
        g[a][b]=(w<g[a][b]?w:g[a][b]);//
    }
    for(int i=0;i<q;i++)
    {
        int c,a,b;
        scanf("%d",&c);
        if(c)
        {
            scanf("%d%d",&a,&b);
            if(fl[a]==0||fl[b]==0)
            {
                printf("ERROR! At path %d to %d\n",a,b);
            }
            else if(g[a][b]>=0x1f1f1f1f)
            {
                printf("No such path\n");
            }
            else
            {
                printf("%d\n",g[a][b]);
            }
        }
        else
        {
            scanf("%d",&a);
            if(fl[a])
            {
                printf("ERROR! At point %d\n",a);
            }
            else
            {
                fl[a]=1;
                //
                for(int j=0;j<n;j++)
                {
                    for(int k=0;k<n;k++)
                    {
                        if(g[j][a]+g[a][k]<g[j][k])
                        {
                            g[j][k]=g[j][a]+g[a][k];
                        }
                    }
                }
            }
        }
    }
    printf("\n");
 }
 return 0;
}
```

3.9 差分约束与最短路

3.9.1 基本原理

```
x1-x2<=0
x1-x5<=-1
x2-x5<=1
x3-x1<=5
x4-x1<=4
x4-x3<=-1
x5-x3<=-3
x5-x4<=-3
```

(u,v)
if(res[v]>res[u]+(u,v))res[v]=res[u]+(u,v),
dis[v]<=dis[u]+(u,v) dis
x[i] i
x[i]-x[j]<=k k (j,i)
Bellman-Ford
SPFA

3.9.2 解题思路

3.9.3 经典题目

1 /

ZOJ 2770

2

n
Ci i,j,k i j k

3

(1). i j k s[j]-s[i-1]>=k,s[i-1]-s[j]<=-k,
(j,i-1) -k

(2). c[i] 0 s[i]-s[i-1]>=0 s[i]-s[i-1]<=c[i] Bellman-Ford SPFA
n 0

4

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

#define EDGE_COUNT 30000
#define PNT_COUNT 2000
#define INF 0x1f1f1f1f

struct edge
{
 int adj;
 int point;
 int weight;
 edge(int a,int p,int w):adj(a),point(p),weight(w){}
 edge():adj(0),point(0),weight(0){}
};

edge e[EDGE_COUNT];
int dis[PNT_COUNT];

int c[1005];

bool bellman_ford(int src,int edge_count,int pnt_count)
{
 memset(dis,0x1f,sizeof(dis));
 dis[src]=0;
 for(int i=1;i<pnt_count;i++)
 {
     for(int j=1;j<=edge_count;j++)
     {
         if(dis[e[j].adj]<INF&&dis[e[j].adj]+e[j].weight<dis[e[j].point])
         {
             dis[e[j].point]=dis[e[j].adj]+e[j].weight;
         }
     }
 }
 for(int i=1;i<=edge_count;i++)
 {
     if(dis[e[i].adj]<INF&&dis[e[i].adj]+e[i].weight<dis[e[i].point])
     {
         return 0;
     }
 }
 return 1;
}

int main()
{
 int n,m;
 while(~scanf("%d%d",&n,&m))
 {
     for(int i=1;i<=n;i++)
     {
         scanf("%d",&c[i]);
     }
     for(int i=1;i<=m;i++)
     {
         int u,v,w;
         scanf("%d%d%d",&u,&v,&w);
         e[i]=edge(v,u-1,-w);//u--v           w         i
     }
     for(int i=1;i<=n;i++)
     {
         e[m+i]=edge(i-1,i,c[i]);//   i            c[i]     m+i
         e[m+n+i]=edge(i,i-1,0);//         0     m+n+i
     }
     if(bellman_ford(n,m+n+n,n+1))printf("%d\n",-dis[0]);//
     else printf("Bad Estimations\n");
 }
 return 0;
}
```

3.10 最大流

3.10.1 基本原理

G=(V,E) V S T
s S t T S T (S,T)

(i,j),
cap(i,j)-flow(i,j),
EK

1. c f;
2. c(i,j)>f(i,j) c(i,j)-f(i,j) m;
3. 1 (i,j) f(i,j)+=m,f(j,i)-=m;

Dinic
1

Dinic EK

1. c f;
2. c(i,j)>f(i,j) i d[i]
3. d[i]==d[j]-1 2.

ISAP Dinic
Dinic 1
1
+1

1. ;
2. ;
3. d[i]==d[j]+1 c(i,j)>f(i,j)
4. 3
3 c(i,j)>f(i,j) d
d 1 3;
num[]
4 5 num[4]=5, num num
0 Gap

3.10.2 解题思路

3.10.3 模板代码

EK

```
int ek(int st,int ed,int src,int tar)//st ed            src tar
{
    int res=0;
    int pre[N];
    int mn[N];
    memset(fl,0,sizeof(fl));
    while(1)
    {
        memset(pre,-1,sizeof(pre));
        mn[src]=INF;
        memset(fl,0,sizeof(fl));
        queue<int>q;
        q.push(src);
        while(!q.empty())
        {
            int t=q.front();
            q.pop();
            if(fl[t])continue;
            fl[t]=1;
            for(int i=st;i<=ed;i++)
            {
                if(!fl[i]&&cap[t][i]-flow[t][i]>0)
                {
                    pre[i]=t;
                    mn[i]=MIN(cap[t][i]-flow[t][i],mn[t]);
                    q.push(i);
                }
            }
            if(pre[tar]!=-1)
            {
                for(int i=tar;pre[i]!=-1;i=pre[i])
                {
                    flow[pre[i]][i]+=mn[tar];
                    flow[i][pre[i]]-=mn[tar];
                }
                break;
            }
        }
        if(pre[tar]==-1)break;
        else res+=mn[tar];
    }
    return res;
}

Dinic
#define INF 0x1f1f1f1f
#define MIN(a,b) ((a)<(b)?(a):(b))
#define MAX(a,b) ((a)>(b)?(a):(b))

#define N 500

int cap[N][N];//
int flow[N][N];//

int lev[N];//
bool vis[N];//

int que[100000];//

//BSF
```

```
//st            ed            src         tar
bool bfs(int st,int ed,int src,int tar)//st      ed      src    tar
{
   int front;//
   int rear;//
   front=rear=0;

   que[front++]=src;
   lev[src]=0;
   memset(vis,0,sizeof(vis));
   vis[src]=1;

   while(rear<front)
   {
      int t=que[rear];
      rear++;

      for(int i=st;i<=ed;i++)
      {
         if(!vis[i]&&cap[t][i]>flow[t][i])
         {
            vis[i]=1;
            lev[i]=lev[t]+1;
            que[front++]=i;
         }
      }
   }
   return lev[tar]<INF;
}

//                      DFS                     DFS
//mn
int dfs(int v,int st,int ed,int tar,int fl)//fl
{
   int ret=0;
   if(v==tar||fl==0)return fl;
   for(int i=st;i<=ed;i++)
   {
      if(fl==0)break;
      if(cap[v][i]>flow[v][i]&&lev[v]+1==lev[i])
      {
         int f=MIN(fl,cap[v][i]-flow[v][i]);//  i
         int tt=dfs(i,st,ed,tar,f);//  i
         if(tt<=0)continue;
         ret+=tt;
         fl-=tt;//       fl
         flow[v][i]+=tt;
         flow[i][v]-=tt;
      }
   }
   return ret;
}

int dinic(int st,int ed,int src,int tar)
{
   int ret=0;
   while(bfs(st,ed,src,tar))//
   {
      int r=dfs(src,st,ed,tar,INF);
      if(r==0)break;
      ret+=r;
   }
   return ret;
}

ISAP
/*      SAP
```

```
 * GAP +
 * GAP:
 *                             O(V)
 * */

const int INF = INT_MAX / 3;
const int MAXN = 20000 + 5;
const int MAXM = 200000 + 5;

struct Edge {
 int u, v;
 int c;
 int next;
 Edge() {}
 Edge(int tu, int tv, int tc, int tn) : u(tu), v(tv), c(tc), next(tn) {}
};

Edge E[MAXM * 3];
// head[]
int nE, head[MAXN], cnt[MAXN], que[MAXN], d[MAXN], low[MAXN], cur[MAXN];

void addEdge(int u, int v, int c, int rc = 0) { // c           rc
 E[nE]= Edge(u, v, c, head[u]);
 head[u] = nE++;
 E[nE]= Edge(v, u, rc, head[v]);
 head[v] = nE++;
}

void initNetwork(int n = MAXN) { // head[]           -1
 memset(head, -1, sizeof(head[0])*n);
 nE = 0;
}

int maxflow(int n, int source, int sink) {
 int *fr = que, *ta = que;;
 for (int i = 0; i < n; ++i) d[i] = n, cnt[i] = 0;
 cnt[n] = n - 1, cnt[0]++, d[sink] = 0;
 *ta++ = sink;
 while (fr < ta) {
     int v = *fr++;
     for (int i = head[v]; i != -1; i = E[i].next) {
         if (d[E[i].v] == n && E[i^1].c > 0) {
             d[E[i].v] = d[v] + 1;
             cnt[n]--;
             cnt[d[E[i].v]]++;
             *ta++ = E[i].v;
         }
     }
 }
 int flow = 0, u = source, top = 0;
 low[0] = INF;
 for (int i = 0; i < n; ++i) cur[i] = head[i];
 while (d[source] < n) { // que      pre
     int &i = cur[u];
     for (; i != -1; i = E[i].next) {
         if (E[i].c > 0 && d[u] == d[E[i].v] + 1) {
             low[top+1] = low[top] < E[i].c ? low[top] : E[i].c;
             que[top+1] = i;
             ++top;
             u = E[i].v;
             break;
         }
     }
     if (i != -1) {
         if (u == sink) {
             int delta = low[top];
             for (int p = 1, i; p <= top; ++p) {
                 i = que[p];
                 E[i].c -= delta;
                 E[i^1].c += delta;
             }
             flow += delta;
             u = source;
             low[0] = INF;
             top = 0;
         }
     } else {
         int old_du = d[u];
         cnt[old_du]--;
         d[u] = n - 1;
         for (int i = head[u]; i != -1; i = E[i].next) {
             if (E[i].c > 0 && d[u] > d[E[i].v]) d[u] = d[E[i].v];
         }
         cnt[++d[u]]++;
         if (d[u] < n) cur[u] = head[u];
         if (u != source) {
             u = E[que[top]].u;
             --top;
         }
         if (cnt[old_du] == 0) break;
     }
 }
 return flow;
}
```

3.10.4 经典题目

3.10.4.1 题目 1

POJ 3084

2

3

1

4

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

int m,n;

#define INF 0x1f1f1f1f
#define MIN(a,b) ((a)<(b)?(a):(b))
#define MAX(a,b) ((a)>(b)?(a):(b))

#define SIZE 30

int flow[SIZE][SIZE];
int cap[SIZE][SIZE];
int lev[SIZE];
int mn[SIZE];

int que[100000];
```

```
//BSF
//st            ed            src         tar
bool bfs(int st,int ed,int src,int tar)
{
   memset(lev,0x1f,sizeof(lev));

   int front;
   int rear;
   front=rear=0;

   que[front++]=src;
   lev[src]=0;

   while(rear<front)
   {
      int t=que[rear];
      rear++;

      for(int i=st;i<=ed;i++)
      {
         if(cap[t][i]>flow[t][i])
         {
            if(lev[t]+1<lev[i])
            {
               lev[i]=lev[t]+1;
               que[front++]=i;
            }
         }
      }
   }
   return lev[tar]<INF;
}

//                      DFS                     DFS
//mn
int dfs(int v,int st,int ed,int src,int tar)
{
   int ret=0;
   if(v==tar)return mn[tar];
   for(int i=st;i<=ed;i++)
   {
      if(mn[v]==0)break;
      if(cap[v][i]>flow[v][i]&&lev[v]+1==lev[i])
      {
         mn[i]=MIN(mn[v],cap[v][i]-flow[v][i]);
         int tt=dfs(i,st,ed,src,tar);
         ret+=tt;
         mn[v]-=tt;//       mn[v]
         flow[v][i]+=tt;
         flow[i][v]-=tt;
      }
   }
   if(ret==0)
   {
      lev[v]=INF;
   }
   return ret;
}

int dinic(int st,int ed,int src,int tar)
{
   int ret=0;
   while(bfs(st,ed,src,tar))//
   {
      memset(mn,0x1f,sizeof(mn));
      int r=dfs(src,st,ed,src,tar);
      if(r==0)break;
      ret+=r;
   }
   return ret;
}

int main()
{
   //printf("%d\n",INF);
   int t;
   scanf("%d",&t);
   while(t--)
   {
      memset(flow,0,sizeof(flow));
      memset(cap,0,sizeof(cap));
      scanf("%d%d",&m,&n);
      for(int i=0;i<m;i++)
      {
         char ch[5];
         int c;
         scanf("%s%d",ch,&c);
         if(strcmp(ch,"I")==0)
         {
            cap[m][i]=INF;
         }
         for(int j=0;j<c;j++)
         {
            int num;
            scanf("%d",&num);
            cap[i][num]=INF;
            if(cap[num][i]<INF)
            {
               cap[num][i]++;
            }
         }
      }
      int res=dinic(0,m,m,n);
      if(res<INF)printf("%d\n",res);
      else printf("PANIC ROOM BREACH\n");
   }
   return 0;
}
```

3.10.4.2 题目 2

1 /

ZOJ 2760

2

G=(V, E) s t s-t 1 <= N <= 100

3

s t floyd dijkstra dis[s][i]+dis[j][t]+g[i][j]==dis[s][t] dis[s][i],dis[j][t],g[i][j] 1 0 s t 0 0.

4

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

#define INF 0x1f1f1f1f
#define MIN(a,b) ((a)<(b)?(a):(b))
#define MAX(a,b) ((a)>(b)?(a):(b))

#define N 500
```

```
int g[N][N];
bool fl[N];
int r[N][N];

int cap[N][N];
int flow[N][N];

int lev[N];
bool vis[N];

int que[100000];

//BSF
//st            ed            src         tar
bool bfs(int st,int ed,int src,int tar)
{
   int front;
   int rear;
   front=rear=0;

   que[front++]=src;
   lev[src]=0;
   memset(vis,0,sizeof(vis));
   vis[src]=1;

   while(rear<front)
   {
      int t=que[rear];
      rear++;

      for(int i=st;i<=ed;i++)
      {
         if(!vis[i]&&cap[t][i]>flow[t][i])
         {
            vis[i]=1;
            lev[i]=lev[t]+1;
            que[front++]=i;
         }
      }
   }
   return lev[tar]<INF;
}

//                      DFS                       DFS
//mn
int dfs(int v,int st,int ed,int tar,int fl)
{
   int ret=0;
   if(v==tar||fl==0)return fl;
   for(int i=st;i<=ed;i++)
   {
      if(fl==0)break;
      if(cap[v][i]>flow[v][i]&&lev[v]+1==lev[i])
      {
         int f=MIN(fl,cap[v][i]-flow[v][i]);
         int tt=dfs(i,st,ed,tar,f);
         if(tt<=0)continue;
         ret+=tt;
         fl-=tt;//       fl
         flow[v][i]+=tt;
         flow[i][v]-=tt;
      }
   }
   return ret;
}

int dinic(int st,int ed,int src,int tar)
{
   int ret=0;
   while(bfs(st,ed,src,tar))//
   {
      int r=dfs(src,st,ed,tar,INF);
      if(r==0)break;
      ret+=r;
   }
   return ret;
}

int main()
{
   int n;
   while(~scanf("%d",&n))
   {
      for(int i=1;i<=n;i++)
      {
         for(int j=1;j<=n;j++)
         {
            scanf("%d",&g[i][j]);
            if(g[i][j]==-1)g[i][j]=INF;
            if(i==j)g[i][j]=0;
            r[i][j]=g[i][j];
         }
      }

      int src,tar;
      scanf("%d%d",&src,&tar);
      src++,tar++;
      if(src==tar)
      {
         printf("inf\n");
         continue;
      }
      for(int k=1;k<=n;k++)
      {
         for(int i=1;i<=n;i++)
         {
            for(int j=1;j<=n;j++)
            {
               if(r[i][k]+r[k][j]<r[i][j])
               {
                  r[i][j]=r[i][k]+r[k][j];
               }
            }
         }
      }


      for(int i=1;i<=n;i++)
      {
         for(int j=1;j<=n;j++)
         {
            if(r[src][i]+r[j][tar]+g[i][j]==r[src][tar]&&g[i][j]<INF)
            {
               cap[i][j]=1;
            }
            else
            {
               cap[i][j]=0;
            }
         }
      }
      memset(flow,0,sizeof(flow));
      int res=dinic(1,n,src,tar);
      printf("%d\n",res);
   }
   return 0;
}
```

3.11 最小费用最大流

3.11.1 基本原理

SPFA EK

3.11.2 模板代码

```
#define INF 0x3f3f3f3f
#define MIN(a,b) ((a)<(b)?(a):(b))

int mat[55][55];
int n,k;

int head[5005];
struct Arc
{
   int next_arc;
   int point;
   int adj;
   int cost;
   int cap;
};
struct Arc arc[25000];

int pre[5005];
int dis[5005];
bool fl[5005];

int max_flow;
int min_cost;

int edge_cnt;

void add(int u,int v,int cst,int cp)
{
   arc[edge_cnt].next_arc=head[u];
   arc[edge_cnt].point=v;
   arc[edge_cnt].adj=u;
   arc[edge_cnt].cost=cst;
   arc[edge_cnt].cap=cp;
   head[u]=edge_cnt;
}

void cost_flow(int src,int tar)
{
   while(1)
   {
      memset(pre,-1,sizeof(pre));
      memset(dis,0x3f,sizeof(dis));
      memset(fl,0,sizeof(fl));

      queue<int>q;
      q.push(src);
      dis[src]=0;

      while(!q.empty())
      {
         int u=q.front();
         q.pop();
         fl[u]=0;

         for(int e=head[u];e!=-1;e=arc[e].next_arc)
         {
            if(arc[e].cap>0&&dis[u]+arc[e].cost<dis[arc[e].point])
            {
               dis[arc[e].point]=dis[u]+arc[e].cost;
               pre[arc[e].point]=e;
               if(!fl[arc[e].point])
               {
                  fl[arc[e].point]=1;
                  q.push(arc[e].point);
               }
            }
         }
      }

      if(pre[tar]==-1)break;

      int min=INF;

      for(int i=tar;pre[i]!=-1;i=arc[pre[i]].adj)
      {
         min=MIN(min,arc[pre[i]].cap);
      }
      for(int i=tar;pre[i]!=-1;i=arc[pre[i]].adj)
      {
         arc[pre[i]].cap-=min;
         arc[pre[i]^1].cap+=min;
      }

      max_flow+=min;
      min_cost+=min*dis[tar];
   }
}
```

3.11.3 经典题目

1 /

POJ 3422

2

N*N

sum sum 0 K sum

3

1

0 1 n*n*2 K

0

4

```
int main()
{
   while(~scanf("%d%d",&n,&k))
   {
      max_flow=0;
      min_cost=0;

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

      for(int i=1;i<=n;i++)
      {
         for(int j=1;j<=n;j++)
         {
            scanf("%d",&mat[i][j]);
         }
      }

      edge_cnt=0;

      for(int i=1;i<=n;i++)
      {
         for(int j=1;j<=n;j++)
         {
            int p3=(i-1)*n+j;
            int p4=(i-1)*n+j+n*n;

            add(p3,p4,-mat[i][j],1);
            edge_cnt++;

            add(p4,p3,mat[i][j],1);
            edge_cnt++;


            add(p3,p4,0,k);
            edge_cnt++;

            add(p4,p3,0,0);
            edge_cnt++;

            if(i<n)
            {
               int p1=(i-1)*n+j;
               int p2=i*n+j;

               add(p1+n*n,p2,0,k);
               edge_cnt++;

               add(p2,p1+n*n,0,0);
               edge_cnt++;

            }
            if(j<n)
            {
               int p1=(i-1)*n+j;
               int p2=(i-1)*n+j+1;

               add(p1+n*n,p2,0,k);
               edge_cnt++;
               add(p2,p1+n*n,0,0);
               edge_cnt++;
           }
         }
      }

      add(0,1,0,k);
      edge_cnt++;

      add(1,0,0,0);
      edge_cnt++;

      edge_cnt++;
      add(n*n*2,n*n*2+1,0,k);
      edge_cnt++;
      add(n*n*2+1,n*n*2,0,0);

      cost_flow(0,2*n*n+1);

      printf("%d\n",-min_cost);
   }
   return 0;
```

```
}
```

3.12 有上下界的最大流

. 2011

3.12.1 基本原理

0

c, b s, t.

1. Vs,Tt,
2. vi, (vi,Vt), vi
3. vi, (Vs,vi), vi
4. (u,v) c(u,v)-b(u,v)
5. (s,t),(t,s) INF 0.
6. Vs Vt

Vs

3.12.2 经典题目

1 /

POJ 2396

2

3

r,q

special judge,

4

```
int main()
{
   int t;
   scanf("%d",&t);
   bool first=1;
   while(t--)
   {
      memset(cap,0,sizeof(cap));
      memset(g1,0,sizeof(g1));
      memset(g2,0x1f,sizeof(g2));
      memset(flow,0,sizeof(flow));
      scanf("%d%d",&m,&n);
      for(int i=0;i<m;i++)//  1                          i         i
      {
         int nu;
         scanf("%d",&nu);

         g1[1][i+2]=MAX(g1[1][i+2],nu);
         g2[1][i+2]=MIN(g2[1][i+2],nu);
      }
      for(int i=0;i<n;i++)//    n       m+n+2
      {
         int nu;
         scanf("%d",&nu);

         g1[m+i+2][m+n+2]=MAX(g1[m+i+2][m+n+2],nu);
         g2[m+i+2][m+n+2]=MIN(g2[m+i+2][m+n+2],nu);
      }

      cap[1][m+n+2]=cap[m+n+2][1]=INF;

      int c;
      scanf("%d",&c);

      for(int i=0;i<c;i++)
      {
         int r,q,v;
         char ch;
         scanf("%d%d %c%d",&r,&q,&ch,&v);
         if(r!=0&&q!=0)
         {
            if(ch=='=')
            {
               g1[r+1][q+m+1]=MAX(g1[r+1][q+m+1],v);
               g2[r+1][q+m+1]=MIN(g2[r+1][q+m+1],v);
            }
            else if(ch=='<')
            {
               g2[r+1][q+m+1]=MIN(g2[r+1][q+m+1],v-1);
            }
            else
            {
               g1[r+1][q+m+1]=MAX(g1[r+1][q+m+1],v+1);
            }
         }
         else
         {
            if(r==0&&q==0)
            {
               if(ch=='=')
               {
                  for(int i1=0;i1<m;i1++)
                  {
                     for(int j1=0;j1<n;j1++)
                     {
                        g1[i1+2][m+2+j1]=MAX(g1[i1+2][m+2+j1],v);
                        g2[i1+2][m+2+j1]=MIN(g2[i1+2][m+2+j1],v);
                     }
      }
   }
   else if(ch=='<')
   {
      for(int i1=0;i1<m;i1++)
      {
         for(int j1=0;j1<n;j1++)
         {
            g2[i1+2][m+2+j1]=MIN(g2[i1+2][m+2+j1],v-1);
         }
      }
   }
   else
   {
      for(int i1=0;i1<m;i1++)
      {
         for(int j1=0;j1<n;j1++)
         {
            g1[i1+2][m+2+j1]=MAX(g1[i1+2][m+2+j1],v+1);
         }
      }
   }
}
else if(r==0)
{
   if(ch=='=')
   {
      for(int i=0;i<m;i++)
      {
         g1[i+2][m+q+1]=MAX(g1[i+2][m+q+1],v);
         g2[i+2][m+q+1]=MIN(g2[i+2][m+q+1],v);
      }
   }
   else if(ch=='<')
   {
      for(int i=0;i<m;i++)
      {
         g2[i+2][m+q+1]=MIN(g2[i+2][m+q+1],v-1);
      }
   }
   else
   {
      for(int i=0;i<m;i++)
      {
         g1[i+2][m+q+1]=MAX(g1[i+2][m+q+1],v+1);
      }
   }
}
else
{
   if(ch=='=')
   {
      for(int i=0;i<n;i++)
      {
         g1[r+1][m+2+i]=MAX(g1[r+1][m+2+i],v);
         g2[r+1][m+2+i]=MIN(g2[r+1][m+2+i],v);
      }
   }
   else if(ch=='<')
   {
      for(int i=0;i<n;i++)
      {
         g2[r+1][m+2+i]=MIN(g2[r+1][m+2+i],v-1);
      }
   }
   else
   {
      for(int i=0;i<n;i++)
      {
```

```
                     g1[r+1][m+2+i]=MAX(g1[r+1][m+2+i],v+1);
                  }
               }
            }
         }
      }

      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            cap[i+2][j+m+2]=g2[i+2][j+m+2]-g1[i+2][j+m+2];//
         }
      }

      for(int i=1;i<=m+n+2;i++)
      {
         int num=0;
         for(int j=1;j<=m+n+2;j++)
         {
            num+=g1[i][j];
         }
         cap[i][m+n+3]=num;//    i                i
         num=0;
         for(int j=1;j<=m+n+2;j++)
         {
            num+=g1[j][i];
         }
         cap[0][i]=num;
      }

      dinic(0,m+n+3,0,m+n+3);//                               0           m+n+3


      if(!first)//                  ~
      {
         printf("\n");
      }
      first=0;

      int jud=1;//
      for(int i=1;i<=m+n+3;i++)
      {
         if(flow[0][i]!=cap[0][i])
         {
            jud=0;
            break;
         }
      }
      if(jud==0)
      {
         printf("IMPOSSIBLE\n");
         continue;
      }

      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            if(j>0)printf(" ");
            printf("%d",flow[i+2][j+m+2]+g1[i+2][j+m+2]);//
         }
         printf("\n");
      }
   }
   return 0;
}
```

3.13 树的最小支配集，最小点覆盖与最大独立集

2012 3

3.13.1 基本原理

1 G = (V, E) V
V V` G
u V` V` V`
V` V` G

2 G = (V, E) V
E V` G
u, v , u V` v V` V`
V` V` G

3 G = V, E V
V` G
u, v ,u v V` u v V` V`
V` V` V` G

G
G
3

1.
(1)

① 1

②

1

( )

2

DFS newpos[i]

i now Select[]

p[i] i greedy ,s[i]

true, i Set[i] i

2.

1

① dp[i][0]: i i

② dp[i][1]: i i i 1

③ dp[i][2]: i i i

dp[i][0] 3

1 i

i

dp[i][0] = 1 + ∑min(dp[u][0], dp[u][1], dp[u][2]) u i

i dp[i][1] = INF

i

i

i

min dp[u][0]-dp[u][1] u

if (i) dp[i][1] = INF;

else dp[i][1] = ∑min(dp[u][0], dp[u][1]) + inc u i

inc

if (∑min(dp[u][0], dp[u][1]) dp[u][0]) inc = 0

else inc = min dp[u][0] – dp[u][1] (u i)

i i i

i i i

dp[i][2] = ∑ dp[u][1] (u i)

dp[i][0] i i

dp[i][1] i i

dp[i][0]. 1

dp[i][0] = 1 + ∑ min(dp[u][0], dp[u][1]) (u i)
dp[i][1], i i
i i

dp[i][1] = ∑ dp[u][0] (u i)

dp[i][0] i
dp[i][1] i
dp[i][0] i
i
1
dp[i][0] = 1 + ∑ dp[u][1] (u i)
dp[i][1], i i

dp[i][1] = ∑ max(dp[u][0], dp[u][1])

3.13.2 模板代码

```
int p[maxn];
bool select[maxn];
int newpos[maxn];
int now;
int n, m;
void DFS(int x)
{
   newpos[now++] = x;
   int k;
   for (k=head[x]; k!=-1; k=edge[k].next)
   {
      if (!select[edge[k].to])
      {
         select[edge[k].to] = true;
         p[edge[k].to] = x;
         DFS(edge[k].to);
      }
   }
}


int greedy()
{
   bool s[maxn] = {0};
   bool set[maxn] = {0};
   int ans = 0;
   int i;
   for (i=n-1; i>=0; i--)
   {
      int t = newpos[i];
      if (!s[t])
      {
         if (!set[p[t]])
         {
            set[p[t]] = true;
            ans++;
         }
         s[t] = true;
         s[p[t]] = true;
         s[p[p[t]]] = true;
      }
   }
   return ans;
}


int greedy()
{
   bool s[maxn] = {0};
   bool set[maxn] = {0};
   int ans = 0;
   int i;
   for (i=n-1; i>=1; i--)
   {
      int t = newpos[i];
      if (!s[t] && !s[p[t]])
      {
         set[p[t]] = true;
         ans++;
         s[t] = true;
         s[p[t]] = true;
      }
   }
   return ans;
}


int greedy()
{
   bool s[maxn] = {0};
   bool set[maxn] = {0};
   int ans = 0;
   int i;
   for (i=n-1; i>=0; i--)
   {
      int t = newpos[i];
      if (!s[t])
      {
         set[t] = true;
         ans++;
         s[t] = true;
         s[p[t]] = true;
      }
   }
   return ans;
}

int main()
{
   /*        */
   memset(select,0,sizeof(select));
   now = 0;
```

```
    select[1] = true;
    p[1] = 1;
    DFS(1);
    printf("%d\n",greedy());
}
```

O m , m = n-1 O(n), O(n)

u p u

```
void DP(int u, int p)
{
    dp[u][2] = 0;
    dp[u][0] = 1;
    bool s = false;
    int sum = 0,inc = INF;
    int k;
    for (k=head[u]; k!=-1; k=edge[k].next)
    {
        int to = edge[k].to;
        if (to == p)
        continue;
        DP(to, u);
        dp[u][0] += min(dp[to][0], min(dp[0][1],dp[to][2]));
        if (dp[to][0] <= dp[to][1])
        {
            sum += dp[to][0];
            s = true;
        }
        else
        {
            sum += dp[to][1];
            inc = min(inc,dp[to][0]-dp[to][1]);
        }
        if (dp[to][1] !=INF && dp[u][2]!=INF)
            dp[u][2] += dp[to][1];
        else dp[u][2] = INF;
    }
    if (inc == INF && !s)
        dp[u][1] = INF;
    else
    {
        dp[u][1] = sum;
        if (!s)
        dp[u][1] += inc;
    }
}
```

```
void DP()
{
    dp[u][0] = 1;
    dp[u][1] = 0;
    int k, to;
    for (k=head[u]; k!=-1; k=edge[k].next)
    {
        to = edge[k].to;
        if (to == p)
        continue;
        DP(to,u);
        dp[u][0] += min(dp[to][0], dp[to][1]);
        dp[u][1] += dp[to][0];
    }
}
```

```
void DP()
{
    dp[u][0] = 1;
    dp[u][1] = 0;
    int k, to;
    for (k=head[u]; k!=-1; k=edge[k].next)
    {
        to = edge[k].to;
        if (to == p)
        continue;
        DP(to, u);
        dp[u][0] += dp[to][1];
        dp[u][1] += max(dp[to][0], dp[to][1]);
    }
}
```

dp[root][0] dp[root][1]

O n .

3.14 二分图最大匹配

2012 3
2011 1

3.14.1 基本原理

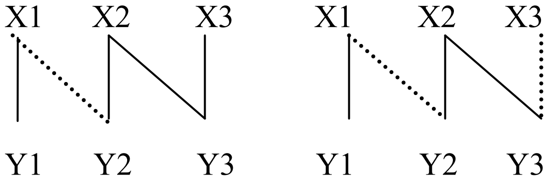
G=<V, E> X Y X Y=V X∩Y=Φ
e={x y} x X y Y G (bipartite graph) <X, E, Y>
X x Y y e E e = {x, y}, G
(complete bipartite graph) |X| = m |Y| = n G Km,n

G G ,

G=<V, E> M⊆E M
M=Φ M
:G
: X(Y) M M M
X- Y- M G
: X(Y)- G
X(Y)-

G=<V, E> M G

M M- M-
: M- M=



1 2

1 [X1,Y2] 2

X2 Y2 X1 Y1 X3 Y3
: X2→Y2→X1→Y1 X2 Y1 M-
Y2 X1 M-
{X2, Y2}, {X1, Y1} {Y2, X1}

X3→Y3

1.
2.
3.
4.
5.
6.
7.

1

3.14.2 解题思路

:

```
while
      do
```

DFS BFS

A
A

3.14.3 模板代码

```
#include <iostream>
#include <string.h>
using namespace std;
int n,k;  //n        k
int V1,V2;        //
 /*                              V1  V2      x  V1   y  V2*/
bool grid[501][501];  //
bool vis[501];      //   V2
int link[501];       //    V2     y    V1           x
int m;  //
bool dfs(int x)
{
    for(int y=1;y<=V2;y++)
        if(grid[x][y] && !vis[y])  //x  y   (   )       y
        {
            vis[y]=true;     //       y
            if(link[y]==0 || dfs(link[y])) //link[y]==0 :    y              M
            {            //find(link[y] :      y
                link[y]=x;   //             M'(  M   M')
                return true;  //
            }
        }
    return false;  //       V1      x
}

void search(void)
{
    for(int x=1;x<=V1;x++)
    {
        memset(vis,false,sizeof(vis)); //
        if(dfs(x))  //  V1       x
            m++;
    }
    return;
}

int main(void)
{
    cin>>n>>k;
    V1=V2=n;

    int x,y;        //   (       )
    for(int i=1;i<=k;i++)
    {
        cin>>x>>y;
        grid[x][y]=true;   //
    }
    search();

    cout<<m<<endl;

    return 0;
}
```

3.14.4 经典题目

3.14.4.1 题目 1

1   : HDU 2063

2   : RPG girls

partner Rabbit

XHD PQK partner Grass linle LL partner PrincessSnow

partner boss partner

Acmer

3 :

4 :

```
#include<stdio.h>
#include<string.h>
#define clr(x)memset(x,0,sizeof(x))
bool g[505][505];
bool v[505];
int l[505];
int n,m;
int find(int k)
{
   int i;
   for(i=1;i<=m;i++)
   {
      if(g[k][i]&&!v[i])
      {
         v[i]=1;  /*     k       i    (i               );
                   *     i         (l[i])     ,
                   *          i              ,                        */
         if(l[i]==0||find(l[i]))
         {
            l[i]=k;
            return 1;
         }
      }
   }
   return 0;
}
int main()
{
   int i,k,p,q,tot;
   while(scanf("%d",&k),k)
   {
      scanf("%d%d",&n,&m);
      clr(g);  clr(l);
      for(i=0;i<k;i++)
      {
         scanf("%d%d",&p,&q);
         g[p][q]=1;
      }
      tot=0;
      for(i=1;i<=n;i++)  //
      {
         clr(v);
         if(find(i))
         tot++;
      }
      printf("%d\n",tot);
```

```
    }
    return 0;
}
```

3.14.4.2 题目 2

1 : POJ 2771 Guardian of Decency

2 : n

3 :

= -

4

```
#include<stdio.h>
#include<string.h>
#define clr(x)memset(x,0,sizeof(x))
struct node
{
    int to,next;
}q[200005];
int head[505];
int tot;
void add(int s,int u) //
{
    q[tot].to=u;
    q[tot].next=head[s];
    head[s]=tot++;
}
struct per
{
    int h;
    char x[3];
    char mu[103];
    char pe[103];
}p[505];
int link[505];
int v[505];
int find(int x)
{
    int i,k;
    for(i=head[x];i;i=q[i].next)
    {
        k=q[i].to;
        if(!v[k])
        {
            v[k]=1;
            if(link[k]==0||find(link[k]))
            {
                link[k]=x;
                return 1;
            }
        }
    }
    return 0;
}
int abs(int x)
{ return x>0?x:-x;  }
int ok(per a,per b)
{
    if(abs(a.h-b.h)>40)
        return 0;
    if(strcmp(a.mu,b.mu))
        return 0;
    if(strcmp(a.pe,b.pe)==0)
        return 0;
    return 1;
}
int main()
{
    int n,t,i,j,sum;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&n);
        tot=1;
        clr(head);
        clr(link);
        for(i=1;i<=n;i++)
            scanf("%d%s%s%s",&p[i].h,p[i].x,p[i].mu,p[i].pe);
        for(i=1;i<=n;i++)
            for(j=1;j<=n;j++)
                if(p[i].x[0]=='F'&&p[j].x[0]=='M'&&ok(p[i],p[j]))
                    add(i,j);
        sum=0;
        for(i=1;i<=n;i++)


        {
            clr(v);
            if(p[i].x[0]=='F'&&find(i))
                sum++;
        }
        printf("%d\n",n-sum);
    }
    return 0;
}
```

3.15 强连通

2012 3

2011 1

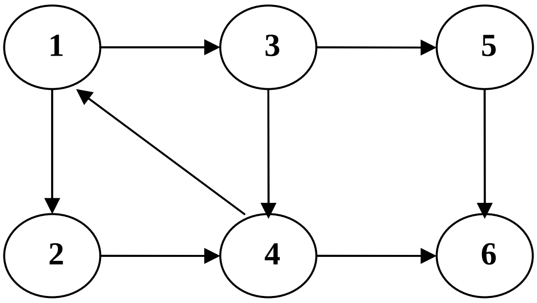
3.15.1 基本原理

1 G G

2 G G ′ v G ′

v G ′ G ′ G

{1,2,3,4} 1,2,3,4 {5},{6}



O(N^2+M)

Kosaraju Tarjan O(N+M)

Tarjan Kosaraju

3.15.2 解题思路

Tarjan

Tarjan

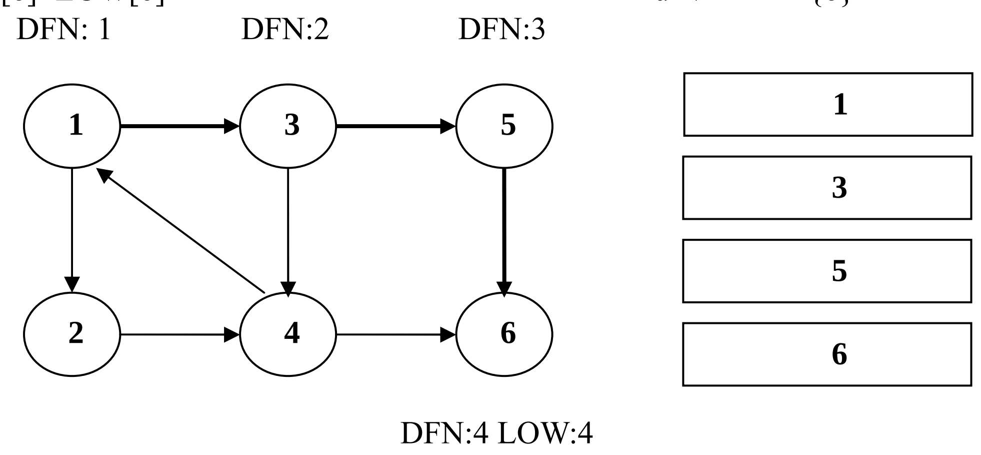
DFN(u) u ( ) Low(u) u u

```
Low(u)=Min
{
   DFN(u),
   Low(v),(u,v)         u  v
   DFN(v),(u,v)                      (       )
}
```

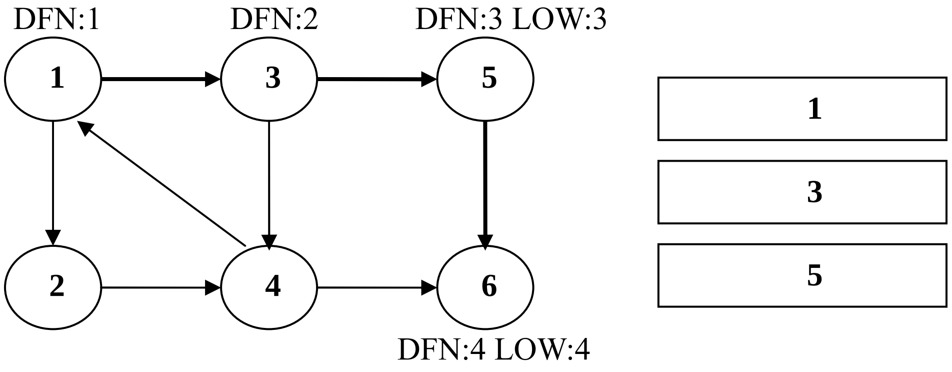
DFN(u)=Low(u) u

```
tarjan(u)
{
    DFN[u]=Low[u]=++Index             //      u             Low
    Stack.push(u)                     //      u
    for each (u, v) in E                  //
        if (v is not visted)                //        v
            tarjan(v)                  //
            Low[u] = min(Low[u], Low[v])
        else if (v in S)                  //        v
            Low[u] = min(Low[u], DFN[v])
    if (DFN[u] == Low[u])               //        u
        repeat
            v = S.pop                 //   v
            print v
        until (u== v)
}
```

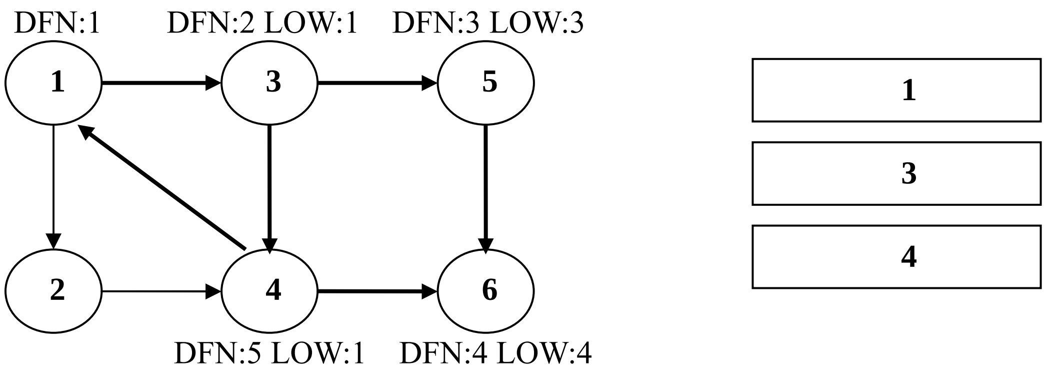
1 DFS u=6 DFN[6]=LOW[6] u=v {6}



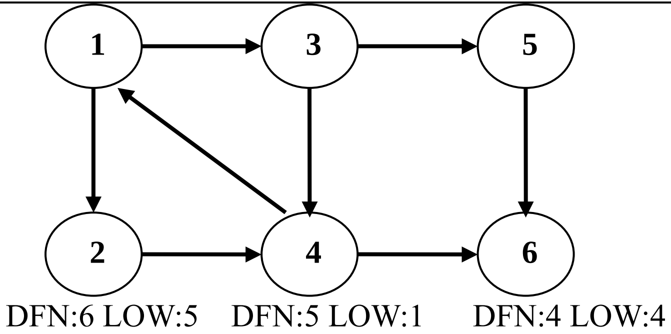
5 DFN[5]=LOW[5] {5}

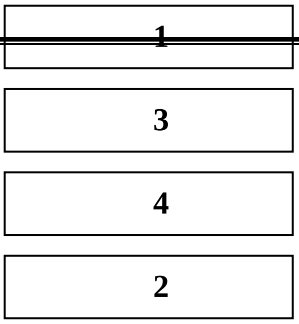


3 4 4 4 1 1 LOW[4]=1 6 (4,6) 3 (3,4) LOW[3]=LOW[4]=1



1 2 (2,4) 4 LOW[2]=DFN[4]=5 1 DFN[1]=LOW[1] {1,3,4,2}





{1,3,4,2},{5},{6}

Tarjan

O(N+M)

3.15.3 模板代码

```
void tarjan(int i)
{
    int j;
    DFN[i]=LOW[i]=++Dindex;
    instack[i]=true;
    Stap[++Stop]=i;
    for (edge *e=V[i];e;e=e->next)
    {
        j=e->t;
        if (!DFN[j])
        {
            tarjan(j);
            if (LOW[j]<LOW[i])
                LOW[i]=LOW[j];
        }
        else if (instack[j] && DFN[j]<LOW[i])
            LOW[i]=DFN[j];
    }

    if (DFN[i]==LOW[i])
    {
        Bcnt++;
        do
        {
            j=Stap[Stop--];
            instack[j]=false;
            Belong[j]=Bcnt;
        }
        while (j!=i);
    }
}
void solve()
{
    int i;
    Stop=Bcnt=Dindex=0;
    memset(DFN,0,sizeof(DFN));
    for (i=1;i<=N;i++)
        if (!DFN[i])
            tarjan(i);
}
```

3.15.4 经典题目

3.15.4.1 题目 1

1 : HDU 1269

2 : n m

3 :

4 ;

```
#include<stdio.h>
#include<string.h>
#define clr(x)memset(x,0,sizeof(x))
const int maxn=100010;
struct node //
{
   int to;
   int next;
}q[100010];
int head[100010];
int tot;
int n,m;
void add(int s,int u)
{
   q[tot].to = u;
   q[tot].next = head[s];
   head[s] = tot++;
}
int dfn[maxn];
int low[maxn];
int stack[maxn];
int ti,sn,top;
bool instack[maxn];
void tarjan(int u)
{
   dfn[u] = low[u] = ++ti;  //
   stack[++top] = u;     //
   instack[u] = true;
   int k, i;
   for (i=head[u]; i; i=q[i].next)
   {
      k = q[i].to;
      if (dfn[k] == 0)   // k         ,k   u      ,  u,k
      {
         tarjan(k);
         if(low[u] > low[k])   //
            low[u] = low[k];
      }
      else if(instack[k] && low[u]>dfn[k])  // k        k  u         u,k
         low[u] = dfn[k];                //
   }
   if(low[u] == dfn[u])        //   u
   {
        sn++;                 //
        do
        {
            k = stack[top--];
            instack[k] = false;
        }
        while(k!=u);
    }
}
int main()
{
    int a,b,i;
    while(scanf("%d %d",&n,&m),n)
    {
        clr(head);
        clr(instack);
        clr(dfn);
        ti = sn = 0;  // ti:            sn:
        tot = 1;
        top = -1;
        while (m--)
        {
            scanf("%d %d",&a,&b);
            add(a,b);
        }
        for (i=1; i<=n; i++)
            if (dfn[i] == 0) //
                tarjan(i);
        if (sn>1)  //                          1
            printf("No\n");
        else printf("Yes\n");
    }
    return 0;
}
```

5

3.15.4.2 题目 2

1 : HDU 2767 Proving Equivalences

2 : n m

3 :

0 0

r g max(r,g)
0

4

```
#include<stdio.h>
#include<string.h>
```

```
#define clr(x)memset(x,0,sizeof(x))
const int maxn=20002;

struct node   //
{
   int to;
   int next;
}q[60000];
int head[maxn];
int tot;
void add(int s,int u)
{
   q[tot].to = u;
   q[tot].next = head[s];
   head[s] = tot++;
}
bool ins[maxn];
int color[maxn];
int dfn[maxn],low[maxn],stack[maxn];
int ti, sn, top;
void tarjan(int u)
{
   dfn[u] = low[u] = ++ti;
   stack[++top] = u;
   ins[u] = true;
   int i, k;
   for (i=head[u]; i; i=q[i].next)
   {
      k = q[i].to;
      if (dfn[k] == 0)
      {
         tarjan(k);
         if (low[k] < low[u])
            low[u] = low[k];
      }
      else if (ins[k] && dfn[k]<low[u])
         low[u] = dfn[k];
   }
   if(dfn[u] == low[u])
   {
      sn++;
      do
      {
         k = stack[top--];
         ins[k] = false;
         color[k] = sn;   //
      }
      while (k!=u);
   }
}
int id[maxn], od[maxn];   //
int main()
{
   int t, i, j, k
   int a, b, n, m;
   scanf("%d",&t);
   while (t--)
   {
      scanf("%d %d",&n,&m);
      tot = 1;
      top = -1;
      sn = ti = 0;
      clr(low);
      clr(dfn);
      clr(ins);
      clr(head);

      while (m--)
      {
         scanf("%d %d",&a,&b);
         add(a,b);
      }
      for (i=1; i<=n; i++)
         if (dfn[i] == 0)
            tarjan(i);
      if (sn == 1)
         printf("0\n");
      else
      {
         clr(id);
         clr(od);
         int in=0,out=0;
         for (i=1; i<=n; i++)
            for(j=head[i]; j; j=q[j].next)
            {
               k = q[j].to;
               if (color[i] != color[k])
               {
                  id[color[k]]++;
                  od[color[i]]++;
               }
            }
         for (i=1; i<=sn; i++)
         {
            if (id[i]==0)
               in++;
            if (od[i]==0)
               out++;
         }
         printf("%d\n",in>out?in:out);
      }
   }
   return 0;
}
```

3.16 重连通

2012 3

2011 1

3.16.1 基本原理

1

2: 1 **(point biconnected)** 1 **(cut point)** **(articulation point)** 1 **(edge biconnected)** 1 **(bridge)** **(articulation edge)**

3: G G ' G ' G ' G ' G ' **(biconnected component)**

Tarjan

3.16.2 解题思路

3.16.2.1 点双连通求解

Tarjan
u
dfn[k] k k
k k >=2 k
k k k
k 2 k
low[son]>= dfn[k] “ k son
w k k k
w k w k son k
father ” k
G u low low[u] u u
Low[u]

```
Low(u)=Min
{
   dfn(u)
   low[v]  v  u
   dfn[v]  v   u           u,v
}
```

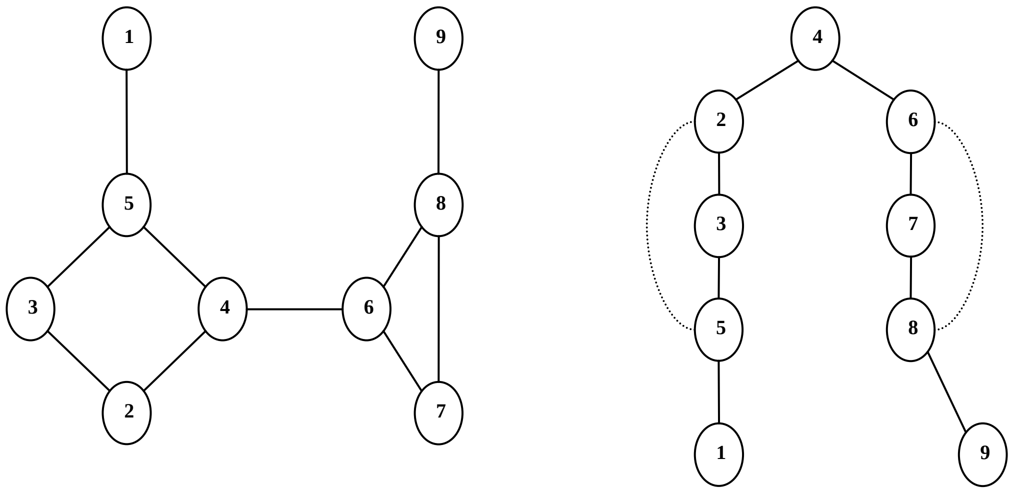
3.16.2.2 边双连通求解

u,v

u,v dfn[u]<low[v]

a 4 DFS dfn[] low[]

b 1 5 4 6 8 9



(a) dfn[] low[] b

Tarjan

low[u]

low[u]=min(low[v],dfn[u]);

dfs

low[u] ,

low[u] dfn(u)<=low[v]

low

low low

dfn

,

3.16.3 模板代码

```
void tarjan(int p,int u)
{
    dfn[u] = low[u] = ++ti;  //
    int i, k;
    int son = 0;    //
    for (i=head[u]; i!=-1; i=edge[i].next)  //
    {
        k = edge[i].to;
        if (k == p)
            continue;
        if (dfn[k] == 0)
        {
            son++;
            tarjan(u,k);
            if (low[k] < low[u])
                low[u] = low[k];
            if((u!=1 && dfn[u]<=low[k]) || (u==1&&son>=2))
                istcc[u]=1;   //          u
        }
        else
            low[u] = min(low[u],dfn[k]);
    }
}



void tarjan(int p,int u)
{
   low[u] = ++ti;
   int k, i;
   for (i=head[u]; i!=-1; i=edge[i].next)
   {
      k = edge[i].to;
      if (k == p) continue;
      if (low[k] == 0)
         tarjan(u,k);
      if (low[k] < low[u])
         low[u] = low[k];
   }
}
```

3.16.4 经典题目

3.16.4.1 题目 1

1   : POJ 1523 SPF

2   :

3   :

4   :

```
#include<stdio.h>
#include<string.h>
#define min(a,b)(a)<(b)?(a):(b)
#define max(a,b)(a)>(b)?(a):(b)
#define clr(x)memset(x,0,sizeof(x))
const int maxn=1010;

struct node  //
{
   int to,next;
}e[10010];
int tot;
int head[maxn];
void add(int s,int u)
{
   e[tot].to = u;
   e[tot].next = head[s];
```

```
    head[s] = tot++;
}
int st,en;
int ti;
int dfn[maxn];
int low[maxn];
int num[maxn];
void dfs(int p, int u)
{
    dfn[u] = low[u] = ++ti;
    int i, k;
    int son = 0;
    for (i=head[u]; i; i=e[i].next)
    {
        k = e[i].to;
        if (k!=p && dfn[k]>0)
            low[u] = min(low[u],dfn[k]);
        else if(dfn[k] == 0)
        {
            son++;
            dfs(u,k);
            if (low[k] < low[u])
                low[u] = low[k];
            if ((u==st && son>=2) || (u!=st && dfn[u]<=low[k]))
                num[u]++;  //
        }
    }
}
int main()
{
    int a,b,i;
    int ca=1;
    while (scanf("%d",&a),a)
    {
        scanf("%d",&b);
        ti = 0;
        tot = 1;
        clr(head);
        st = 1005;
        en = 0;
        st = min(st,a);
        st = min(st,b);
        en = max(en,a);
        en = max(en,b);
        add(a,b);
        add(b,a);
        while (scanf("%d",&a),a)
        {
            scanf("%d",&b);
            st = min(st,a);
            st = min(st,b);
            en = max(en,a);
            en = max(en,b);
            add(a,b);
            add(b,a);
        }
        clr(dfn); clr(low);
        clr(num);
        dfs(0,st);
        printf("Network #%d\n",ca++);
        bool flag = true;
        for (i=st; i<=en; i++)
            if(num[i])
            {
                flag=false;
                printf("  SPF node %d leaves %d subnets\n",i,num[i]+1);
            }
        if (flag)
        printf("  No SPF nodes\n");
        printf("\n");
    }
    return 0;
}
```

3.16.4.2 题目 2

1 : POJ 3352 Road Construction

2 : N

3 :

low

= 1 +1 / 2

4

```
#include<stdio.h>
#include<string.h>
#define clr(x)memset(x,0,sizeof(x))
const int maxn=5005;

struct node  //
{
    int to,next;
}e[10005];
int tot;
int head[maxn];
void add(int s,int u)
{
    e[tot].to=u;
    e[tot].next=head[s];
    head[s]=tot++;
}

int ti,top,sn;
int low[maxn];
void tarjan(int p,int u)
{
    low[u] = ++ti;
    int k,i;
    for (i=head[u]; i; i=e[i].next)
```

```
   {
      k = e[i].to;
      if (k == p)
         continue;
      if (low[k] == 0)
         tarjan(u,k);
      if (low[k] < low[u])
         low[u] = low[k];
   }
}

int main()
{
   int n, m, re;
   int i, j, k, a, b;
   int degree[maxn];
   scanf("%d %d",&n,&m);
   ti = 0;
   tot = 1;
   clr(head);
   clr(low);
   while (m--)
   {
      scanf("%d %d",&a,&b);
      a--; b--;
      add(a, b);
      add(b, a);
   }
   tarjan(0,0);
   clr(degree);
   for (i=0; i<n; i++)
      for (j=head[i]; j; j=e[j].next)
      {
         k = e[j].to;
         if (low[k] != low[i])
            degree[low[i]]++;
      }
   re = 0;
   for (i=1; i<=ti; i++)
      if (degree[i] == 1)
         re++;
   printf("%d\n",(re+1)/2);
   return 0;
}
```

3.17 2-SAT

2012 3
2011 1

3.17.1 基本原理

1 x “ ” “ ” “ ” “ ” x
x

x p(x)

SAT

x1 x2 (x1 x2) p(x1)= p(x2)=

x1 x2 (x1 x2) p(x1),p(x2)

X= x1,x2..,xn x1,x2 ..,xn
“ ” “ ” m C1,C2,..,Cm, C1 C2 ,.., Cm

“ ” “ ”

2: C1,C2,..,Cm, max{|Ci|}=k(1 i m), k
k-SAT
k>2 k-SAT NP k=2

3.17.2 解题思路

2-SAT 2-SAT
Poi 0106 Peaceful Commission [ ]
n 2

1 2n 2a-1 2a a

n m m
1 ≤ n ≤ 8000 0 ≤ m ≤ 20000

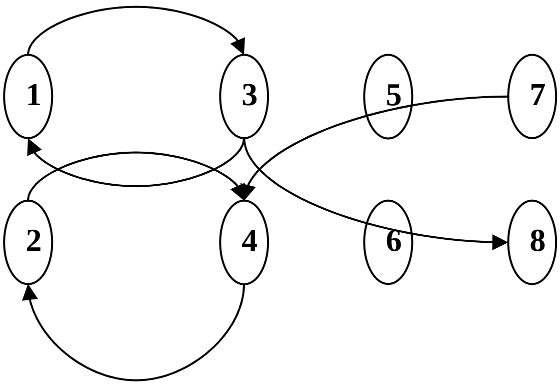
3	2	1
1	3	4
2	4	5

n i Ai, Ai'
n Ai,
Ai'
Ai Ai'

Ai Aj Ai Aj` Aj
Ai`

Ai → Aj`
Aj → Ai`

4 1 4 2 3 7 3



1

1
3 2
5 6
8 4 7

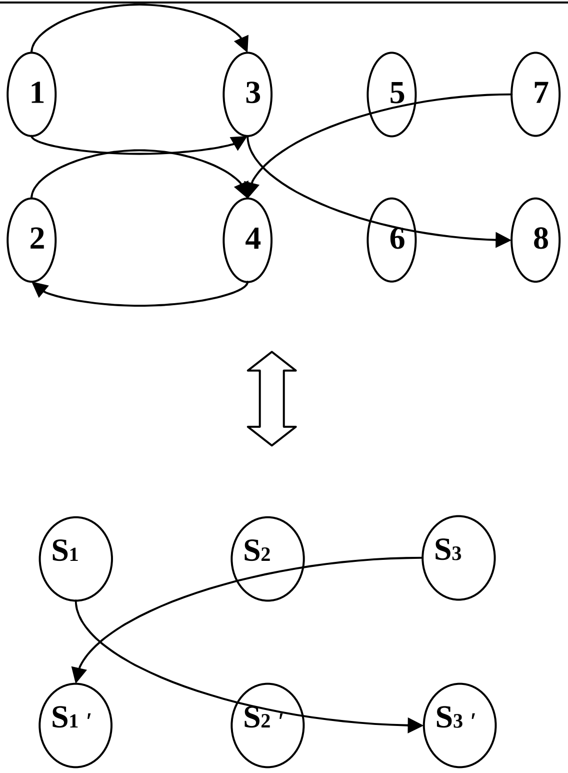
Ai Ai
1
Ai, Ai ‘ 1
1

Ai,Ai' Ai,
Ai'
O(nm)

1 1 3 1 3 2
4

Ai→Aj Ai Si Aj Sj Si ≠ Sj

Si → Sj



Ai, Ai'

2-SAT

2-SAT 11

A[x]

NOT A[x]

A[x] AND A[y]

A[x] AND NOT A[y]

A[x] OR A[y]

A[x] OR NOT A[y]

NOT (A[x] AND A[y])

NOT (A[x] OR A[y])

A[x] XOR A[y]

NOT (A[x] XOR A[y])

A[x] XOR NOT A[y]

And 1 ~x->y, ~y->x (1)

And 0 y->~x , x->~y(0)

OR 1 ~x->y , ~y->x(1)

OR 0 x->~x , y->~y(0)

XOR 1 x->~y , ~x->y , ~y->x , y -> ~x (0 1)

XOR 0 x->y , ~x->~y , y->x ~y->~x(1 0)

tarjan

x ~x +tarjan

3.17.3 模板代码

Tarjan

3.17.4 经典题目

3.17.4.1 题目 1

1 POJ 3207 Ikki's Story IV - Panda's Trick

2 n n m

3 : i , i+m i, j i j+m j i+m i+m j j+m i

i -> j+m
j -> i+m
i+m -> j
j+m -> i

4

```
#include<stdio.h>
#include<string.h>
#define maxn 1100
#define clr(x)memset(x,0,sizeof(x))
#define min(a,b)(a)<(b)?(a):(b)
#define max(a,b)(a)>(b)?(a):(b)
struct Edg  //
{
   int u, v;
}e[1100];
struct node
{
   int to, next;
}edge[1000000];
int tot;
int head[maxn];
void add(int s,int t)
{
   edge[tot].to = t;
   edge[tot].next = head[s];
   head[s] = tot++;
}

int dfn[maxn];
int low[maxn];
int sta[maxn];
int ins[maxn];
int col[maxn];  //
int top,sn, ti;
```

```
void tarjan(int u)
{
    dfn[u] = low[u] = ++ti;
    sta[++top] = u;
    ins[u] = 1;
    int i, k;
    for (i=head[u]; i; i=edge[i].next)
    {
        k = edge[i].to;
        if (dfn[k] == 0)
        {
            tarjan(k);
            low[u] = min(low[u], low[k]);
        }
        else if (ins[k])
            low[u] = min(low[u],dfn[k]);
    }
    if (dfn[u] == low[u])
    {
        sn++;
        do
        {
            k = sta[top--];
            ins[k] = 0;
            col[k] = sn;
        }while (k!=u);
    }
}
int main()
{
    int n,m;
    int i,j;
    while(scanf("%d %d",&n,&m)!=EOF)
    {
        for (i=0; i<m; i++)
        {
            scanf("%d %d",&e[i].u,&e[i].v);
            if (e[i].u > e[i].v)
            {
                int tmp = e[i].u;
                    e[i].u = e[i].v;
                    e[i].v = tmp;
            }
        }
        tot=1;
        clr(head);
        for (i=0; i<m; i++)
            for (j=i+1; j<m; j++)
                if (e[i].u<e[j].v && e[i].u>e[j].u && e[i].v>e[j].v
                  ||e[i].v>e[j].u && e[i].v<e[j].v && e[i].u<e[j].u)
                {
                    add(i, j+m);
                    add(j, i+m);
                    add(j+m, i);
                    add(i+m, j);
                }
        sn = top = ti = 0;
        clr(dfn);  clr(low);
        clr(ins);  clr(col);
```

```
      for (i=0; i<2*m; i++)
         if (dfn[i] == 0)
            tarjan(i);
      for (i=0; i<m; i++)
         if (col[i] == col[i+m])  //
            break;
      if(i == m)
         printf("panda is telling the truth...\n");
      else
         printf("the evil panda is lying again\n");
   }
   return 0;
}
```

3.17.4.2 题目 2

1 : POJ 3683 Priest John's Busiest Day

2 : n , s,
t las, [s,s+las] [t-las,t] ,

3 : 2-SAT
:

i = [s,s+las],i +n = [t-las,t]

i j i —> j + n
i j + n i —> j
i + n j i + n —> j + n
i + n j + n i + n —> j
i i+n

4 :

```
#include<stdio.h>
#include<string.h>
#define min(a,b)(a)<(b)?(a):(b)
#define max(a,b)(a)>(b)?(a):(b)
#define clr(x)memset(x,0,sizeof(x))
#define maxn 2100
#define maxm 3000000

struct node   //
{
   int from, to, next;
}e[maxm],sed[maxm];
int head[maxn];
int sorh[maxn];
int tot;
int tt;

void add(int s,int t)
{
   e[tot].from = s;
   e[tot].to = t;
   e[tot].next = head[s];
   head[s] = tot++;
}
```

```
void add2(int s,int t)
{
   sed[tt].to = t;
   sed[tt].next = sorh[s];
   sorh[s] = tt++;
}
int ti,sn,top,n;
int low[maxn];
int dfn[maxn];
int ins[maxn];
int sta[maxn];
int col[maxn];
int sco[maxn];
int ct[maxn];
int q[maxn];
int ind[maxn];  //
int res[maxn];

void tarjan(int u)
{
   dfn[u] = low[u] = ++ti;
   ins[u] = 1;
   sta[++top] = u;
   int i, k;
   for (i=head[u]; i; i=e[i].next)
   {
      k = e[i].to;
      if (dfn[k] == 0)
      {
         tarjan(k);
         low[u] = min(low[u],low[k]);
      }
      else if (ins[k])
         low[u] = min(low[u],dfn[k]);
   }
   if (dfn[u] == low[u])
   {
      sn++;
      do
      {
         k = sta[top--];
         ins[k] = 0;
         sco[k] = sn; //
      }while(k!=u);
   }
}
struct edge
{
   int s1,e1;
   int s2,e2;
}p[maxn];
char st[22],en[22];
int main()
{
   scanf("%d",&n);
   int i, j, k, las;
   int front, rear;
   for (i=0; i<n; i++)
   {
        scanf("%s %s %d",st,en,&las);
        p[i].s1 = ((st[0]-'0')*10+st[1]-'0')*60+(st[3]-'0')*10+st[4]-'0';
        p[i].e1 = p[i].s1+las;
        p[i].e2 = ((en[0]-'0')*10+en[1]-'0')*60+(en[3]-'0')*10+en[4]-'0';
        p[i].s2 = p[i].e2-las;
    }
    clr(head);
    tot = 1;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
        {
            if (i == j)
                continue;
            if (p[i].s1<p[j].e1 && p[j].s1<p[i].e1)
                add(i,j+n);
            if (p[i].s1<p[j].e2 && p[j].s2<p[i].e1)
                add(i,j);
            if (p[i].s2<p[j].e1 && p[j].s1<p[i].e2)
                add(i+n,j+n);
            if (p[i].s2<p[j].e2 && p[j].s2<p[i].e2)
                add(i+n,j);
        }
        ti = sn = top = 0;
        clr(sco);  clr(dfn);
        clr(low);  clr(ins);
        for (i=0; i<2*n; i++)
            if (!dfn[i])
                tarjan(i);
        int flag=0;
        for (i=0; i<n; i++)
        {
            if (sco[i] == sco[i+n])
            flag = 1;
            ct[sco[i]] = sco[i+n];
            ct[sco[i+n]] = sco[i];
        }
        if(flag)
            goto loop;
        tt = 1;
        clr(sorh);
        clr(ind);
        clr(col);
        for (i=1; i<tot; i++)
            if (sco[e[i].from] != sco[e[i].to])  //
            {
                add2(sco[e[i].to],sco[e[i].from]);
                ind[sco[e[i].from]]++;
            }
        front=0, rear=0;
        for (i=1; i<=sn; i++) //
            if (ind[i] == 0)
                q[rear++] = i;
        while (front<rear)
        {
            int x = q[front++];
            if (col[x] == 0)
            {
                col[x] = 1;
                col[ct[x]] = -1;
```

```
          }
          for (i=sorh[x]; i; i=sed[i].next)
          {
             k = sed[i].to;
             if (--ind[k] == 0)
                q[rear++] = k;
          }
       }
       clr(res);
       for (i=0; i<2*n; i++)
          if (col[sco[i]] == 1)
             res[i] = 1;
loop:   if(flag)
          printf("NO\n");
       else
       {
          printf("YES\n");
          for (i=0; i<n; i++)
          {
             if (res[i])
                printf("%02d:%02d%02d:%02d\n",p[i].s1/60,
                       p[i].s1%60,p[i].e1/60,p[i].e1%60);
             else
                printf("%02d:%02d%02d:%02d\n",p[i].s2/60,
                       p[i].s2%60,p[i].e2/60,p[i].e2%60);
          }
       }
   return 0;
}
```