[报告] H

[source] http://202.114.18.202:8080/judge/contest/view.action?cid=6147#problem/H

[Description]

给定一个有向图，每条边的容量是 1，可以将边反向，然后问源点到汇点的最大流量是多少，并输出需要反向的边的序号。

[方法]

我们可以看出，相当于将原图化为无向图，求源点到汇点的最大流，然后看如果原来方向的边和流量方向相同，就不反向，否则边反向。

[Code]

因为这道题是比赛快结束时候做的，没有好的模板，然后用了 **notonlysuccess** 大神的模板

```
#include "cstdlib"

#include "cctype"

#include "cstring"

#include "cstdio"

#include "cmath"

#include "algorithm"

#include "vector"

#include "string"

#include "iostream"

#include "sstream"

#include "set"
```

```
#include "queue"

#include "stack"

#include "fstream"

#include "iomanip"

#include "bitset"

#include "list"

#include "ctime"

using namespace std;


typedef long long LL;

typedef unsigned long long ULL;

#define CC(m,what)        memset(m,what,sizeof(m))

#define FOR(i,a,b)        for( int i = (a) ; i < (b) ; i ++ )

#define FF(i,a)       for( int i = 0 ; i < (a) ; i ++ )

#define FFD(i,a)      for( int i = (a)-1 ; i >= 0 ; i --)

#define SS(a)         scanf("%d",&a)

#define LL(a)         ((a)<<1)

#define RR(a)         (((a)<<1)+1)

#define SZ(a)         ((int)a.size())

#define PP(n,m,a)     puts("---");FF(i,n){FF(j,m)cout   <<   a[i][j]   <<   '

';puts("");}

const double eps = 1e-11;
const double Pi = acos(-1.0);

#define read              freopen("in.txt","r",stdin)
#define write             freopen("out.txt","w",stdout)

#define two(x)            ((LL)1<<(x))
#define include(a,b)          (((a)&(b))==(b))
template<class T> inline T countbit(T n) {return
n?1+countbit(n&(n-1)):0;}
template<class T> inline T sqr(T a)    {return a*a;}
template<class T> inline void checkmin(T &a,T b)   {if(a == -1 || a > b)a
= b;}
template<class T> inline void checkmax(T &a,T b)  {if(a < b)   a = b;}

int dx[] = {-1,0,1,0};//up Right down Left
int dy[] = {0,1,0,-1};
#define N 310
#define M 1010
int maze[M][M];
int gap[M],dis[M],pre[M],cur[M];
int sap(int s,int t,int nodenum) {
    CC(cur,0);CC(dis,0);CC(gap,0);
    int u = pre[s] = s,maxflow = 0,aug = -1;
    gap[0] = nodenum;
    while(dis[s] < nodenum) {
loop:       FOR(v,cur[u],nodenum) if(maze[u][v] && dis[u] == dis[v] +
1) {
            checkmin(aug,maze[u][v]);
            pre[v] = u;
            u = cur[u] = v;
            if(v == t) {
                maxflow += aug;
                for(u = pre[u];v != s;v = u,u = pre[u]) {
                    maze[u][v] -= aug;
                    maze[v][u] += aug;
                }
                aug = -1;
            }
            goto loop;
        }
        int mindis= nodenum-1;
        FF(v,nodenum) if(maze[u][v] && mindis> dis[v]) {
            cur[u] = v;
            mindis= dis[v];
        }
        if((--gap[dis[u]])== 0)break;
        gap[dis[u] = mindis+1] ++;
        u = pre[u];
    }
    return maxflow;
}
struct E
{
    int x,y;
}tt[M];
int main(void)
{
    int n,m,x,y,st,en;
    scanf("%d%d",&n,&m);
    for(int i=1;i<=m;i++)
    {
        scanf("%d%d",&x,&y);
        maze[x][y]=maze[y][x]=1;
        tt[i].x=x;
        tt[i].y=y;
    }
    scanf("%d%d",&st,&en);
    int ans=sap(st,en,n+1);
    printf("%d\n",ans);
    int sum=0;
    for(int i=1;i<=m;i++)
    {
        if(maze[tt[i].y][tt[i].x]==0)
        {
            sum++;
        }
    }
    printf("%d\n",sum);
    for(int i=1;i<=m;i++)
    {
        if(maze[tt[i].y][tt[i].x]==0)
        {
            printf("%d\n",i);
        }
    }
    return 0;
}
```