可以输入以下数据进行验证。第一行有两个数 n 和 m，n 表示有 n 个顶点，m 表示有 m 条边。接下来 m 行，每行形如“a b”表示顶点 a 和顶点 b 之间有边。

```
6 7
1 4
1 3
4 2
3 2
2 5
2 6
5 6
```

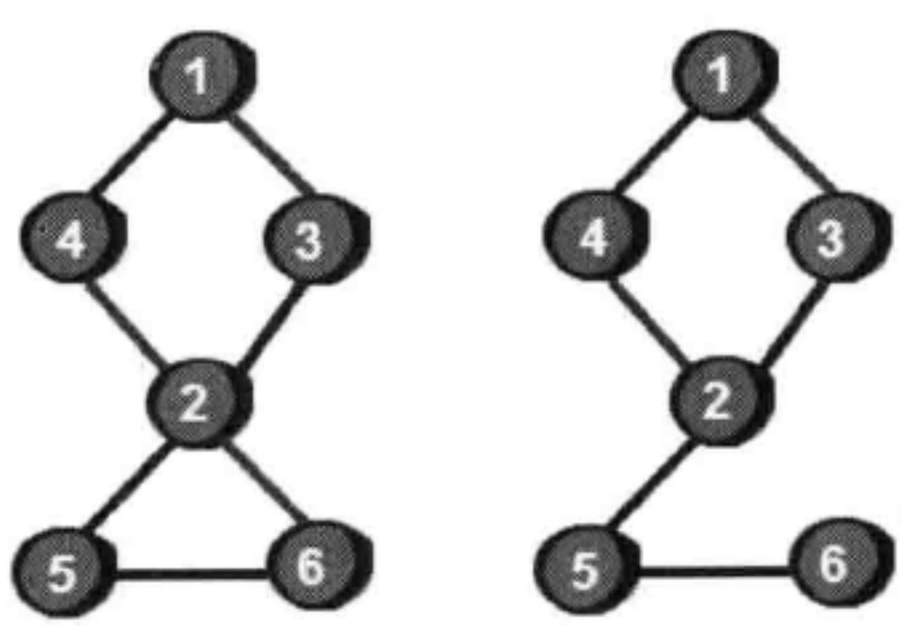
运行结果是：

```
2
```

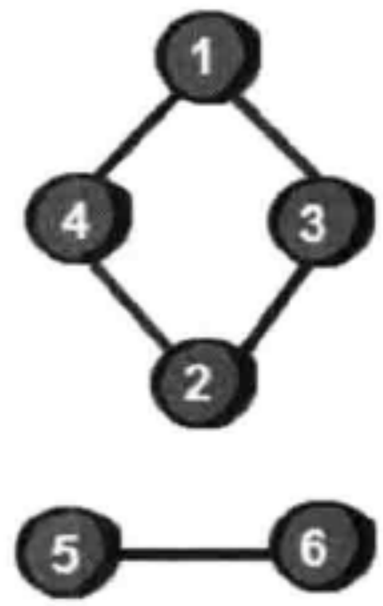
细心的同学会发现，上面的代码是用的邻接矩阵来存储图，这显然是不对的，因为这样无论如何时间复杂度都会在 $O(N^2)$，因为边的处理就需要 N^2 的时间。这里这样写是为了突出割点算法部分，实际应用中需要改为使用邻接表来存储，这样整个算法的时间复杂度是 $O(N+M)$。

第 4 节　关键道路——图的割边

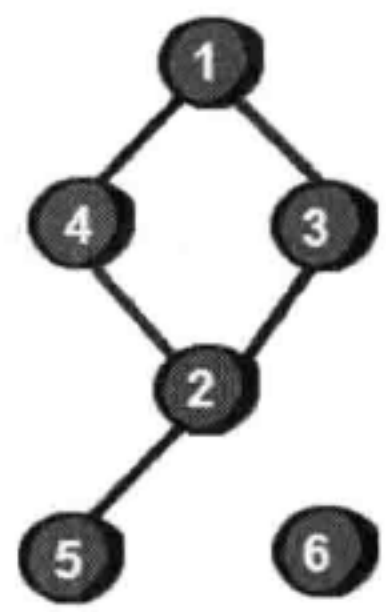
上一节我们解决了如何求割点，还有一种问题是如何求割边（也称为桥），即在一个无向连通图中，如果删除某条边后，图不再连通。下图中左图不存在割边，而右图有两条割边分别是 2-5 和 5-6。



很明显，将 2-5 这条边删除后图被分割成了两个子图，如下。



同理删除 5-6 这条边后图也被分割成了两个子图，如下。



那么如何求割边呢？只需要将求割点的算法修改一个符号就可以。只需将 low[v]>=num[u]改为 low[v]>num[u]，取消一个等于号即可。为什么呢？low[v]>=num[u]代表的是点 v 是不可能在不经过父亲结点 u 而回到祖先（包括父亲）的，所以顶点 u 是割点。如果 low[v]和 num[u]相等则表示还可以回到父亲，而 low[v]>num[u]则表示连父亲都回不到了。倘若顶点 v 不能回到祖先，也没有另外一条路能回到父亲，那么 u-v 这条边就是割边，代码实现如下，请注意输出部分。

```
#include <stdio.h>
int n,m,e[9][9],root;
int num[9],low[9],index;
int min(int a,int b)
{
    return a < b ? a : b;
}
void dfs(int cur,int father)
{
    int i,j;

    index++;
```

```
    num[cur]=index;
    low[cur]=index;
    for(i=1;i <= n;i++)
    {
        if(e[cur][i]==1)
        {
            if(num[i]==0)
            {
                dfs(i,cur);
                low[cur]=min(low[i],low[cur]);
                if(low[i] > num[cur])
                    printf("%d-%d\n",cur,i);
            }
            else if(i != father)
            {
                low[cur]=min(low[cur],num[i]);
            }
        }
    }
}
int main()
{
    int i,j,x,y;
    scanf("%d %d",&n,&m);
    for(i=1;i <= n;i++)
        for(j=1;j <= n;j++)
            e[i][j]=0;

    for(i=1;i <= m;i++)
    {
        scanf("%d %d",&x,&y);
        e[x][y]=1;
        e[y][x]=1;
    }
    root=1;
    dfs(1,root);

    getchar();getchar();
    return 0;
}
```

可以输入以下数据进行验证。第一行有两个数 n 和 m。n 表示有 n 个顶点，m 表示有 m 条边。接下来 m 行，每行形如“a b”表示顶点 a 和顶点 b 之间有边。

```
6 6
1 4
1 3
4 2
3 2
2 5
5 6
```

运行结果是：

```
5-6
2-5
```

同割点的实现一样，这里也是用的邻接矩阵来存储图的，实际应用中需要改为使用邻接表来存储，否则这个算法就不是 $O(N+M)$了，而至少是 $O(N^2)$。如果这样的话，这个算法就没有意义了。因为你完全可以尝试依次删除每一条边，然后用深度优先搜索或者广度优先搜索去检查图是否依然连通。如果删除一条边后导致图不再连通，那么刚才删除的边就是割边。这种方法的时间复杂度是 $O(M(N+M))$。可见一个算法要选择合适的数据结构是非常重要的。

割点和割边算法也是由 Robert E. Tarjan 发明的，不得不说这位同学真是神犇啊！

第 5 节　我要做月老——二分图最大匹配