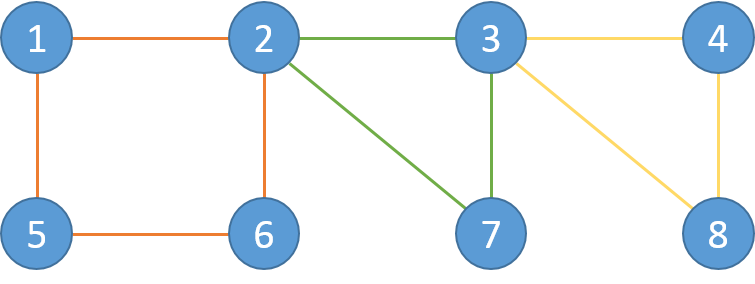
Fleury算法图解

**注意结合代码实现来看：**

在这里我们再来一个有3层的例子：



在这个例子中：

L1: 1-2-6-5-1

L2: 2-3-7-2

L3: 3-4-8-3

第一步时我们将L1压入栈S，同时我们用一个数组Path来记录我们出栈的顺序：

S: [1 2 6 5 1]

Path:

然后出栈到节点2时我们发现了2有其他路径，于是我们把2的另一条路径加入：

S: 1 [2 3 7 2]

Path: 1 5 6

此时L2已经走完，然后再开始弹出元素，直到我们发现3有其他路径，同样压入栈：

S: 1 2 [3 4 8 3]

Path: 1 5 6 2 7

之后依次弹出剩下的元素：

S:

Path: 1 5 6 2 7 3 8 4 3 2 1

此时的Path就正好是我们需要的欧拉路径。

小Ho：原来这样就能求出欧拉路，真是挺巧妙的。

小Hi：而且这个算法在实现时也有很巧妙的方法。因为DFS本身就是一个入栈出栈的过程，所以我们直接利用DFS的性质来实现栈，其伪代码如下：

**DFS(u):**

**While (u存在未被删除的边e(u,v))**

**删除边e(u,v)**

**DFS(v)**

**End**

**PathSize ← PathSize + 1**

**Path[ PathSize ] ← u**