

学生实验报告



课程名称\_\_\_数据结构与算法 \_

题目名称\_\_\_一笔画\_\_

小组成员\_\_ 32001272-徐彬涵 \_\_\_

（学号-姓名）

\_ 32001269-徐若萱 \_

\_ (互评版本请不要填写本行内容) \_

任课教师\_\_ \_

2021 年12月 5 日

# 零.完成情况

**PINTIA评测成绩：**



**成员分工及贡献情况：**

5分 徐彬涵：编写代码测试样例

5分 徐若萱：编写报告

# 一. 问题描述

# 题目：一笔画

小丁最近迷恋上一个游戏，传说中的“一笔画”游戏。

那么什么是一笔画？如下图，顾名思义就是一笔可以完成的图。一笔画最基本的要求是在画图的过程中，笔不能离开纸，且笔所画过的线不能重复，最后画完所有的线便算完成。



虽然小丁喜欢玩这个游戏，但有时候花费半天也找不到答案。小丁听说写一个计算机程序便能判断是否可以一笔画图，所以他希望善良可爱的你来帮帮他的忙。

快来帮帮弱小，可怜，又无助的小丁。

**输入格式:**

给出图中的节点数N（1<=N<=1000,编号1-N）和边数M；随后M行给出存在边的两个节点的编号。

**输出格式:**

能够一笔画的图输出Y，否则输出N。

**输入样例1:**

3 2

1 2

2 3

**输出样例1:**

Y

**输入样例2:**

4 3

1 2

1 3

1 4

**输出样例2:**

N

# 二.数据结构定义与分析

**分析问题**，根据分析写出数据结构定义

建立数组统计每个点连接的边数，建立二维数组来判断两点之间是否有线相连

# 三、算法分析

写出算法的主要思想

首先利用DFS来判断所有的点是否连通，如果不只一个点并且不连通则不能完成一笔画

之后利用统计每个点连接的变数来计算有奇数条边的点的数量，如果有奇数条边的点的数量不为0或2则不能完成一笔画

# 四. 算法的时间复杂度分析

分析算法的时间复杂度以及分析过程

算法可能存在的问题，若有，请写出解决原因与方法。

该算法主要依靠一个DFS算法和for循环，DFS中对每一个点进行遍历，时间复杂度为O(n^2),两个for循环为O(m+n)

# 五、测试用例的选择原因以及测试结果

写出测试用例，详细阐述每个测试用例的选择原因，测试结果可以截图，（注意截图不要出现姓名、学号等信息）

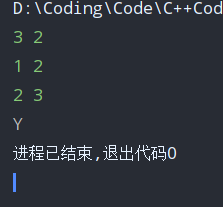
测试样例

3 2

1 2

2 3

Y



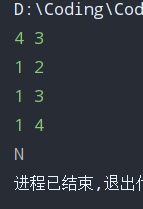
4 3

1 2

1 3

1 4

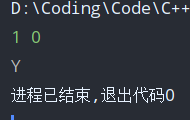
N



只有一个点，无边

1 0

Y



非连通图

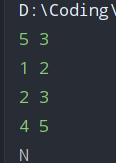
5 3

1 2

2 3

4 5

N



0个奇数条边的点

4 4

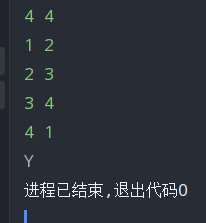
1 2

2 3

3 4

4 1

Y



2个奇数条边的点

4 5

1 2

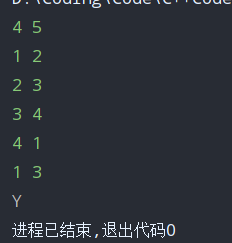
2 3

3 4

4 1

1 3

Y



# 六、存在问题及可能原因

暂未发现问题

# 七、附源代码以及代码相应注释

源代码

1. #include <stdio.h>
2. #include <string.h>
3. #define M 2000*//定义最大节点数*
4. int key;*//统计边数为奇数的点个数*
5. int Graph[M];*//统计每个点连接的边数*
6. int Edge[M][M];*//判断两个点是否连通的数组*
7. int visit[M];*//标记访问过的点*
8. int n, m;
9. *//判断是否连通*
10. void Connect(int x){
11. visit[x] = 1;
12. for(int i = 1; i <= n; ++i){
13. if(Edge[x][i] && !visit[i])
14. Connect(i);*//运用DFS遍历每一个点，经过的点就标记*
15. }
16. }
17. *//判断是否能完成一笔画*
18. bool judge(){
19. key = 0;
20. memset(Graph,0,sizeof(Graph));
21. memset(Edge,0,sizeof(Edge));
22. int x, y;
23. for(int i = 1; i <= m; ++i){
24. scanf("%d %d", &x, &y);
25. Edge[x][y] = 1;
26. Edge[y][x] = 1;
27. Graph[x]++;
28. Graph[y]++;
29. }
30. memset(visit,0,sizeof(visit));
31. Connect(1);*//从第一个点开始遍历*
32. for(int i = 1; i <= n; ++i){
33. if(!visit[i]) return false;*//如果有点没有被访问就证明不是连通图，直接返回false*
34. if(Graph[i] % 2 != 0)
35. key++;
36. }
37. if(key == 0 || key == 2)*//如果边数为奇数的点个数不是0或2则证明无法完成一笔画*
38. return true;
39. else
40. return false;
41. }
42. int main() {
43. scanf("%d %d", &n, &m);
44. if(judge())
45. printf("Y");
46. else
47. printf("N");
48. }

# 八、实验创新点或合理化建议

本次实验的创新之处以及是否可以有更好的建议使得题目更加有实用意义等。

可以将题目扩展，如果当前图像一笔画无法完成，则至少需要几笔画