图的深度优先遍历和广度优先遍历 实验报告

**谭源**

**(信息科学与工程学院，2019计算机类3班，甘肃 兰州 730000)**

1. 测试样例

* 如图所示图（指导中给出的）

图片包含 游戏机, 电脑

描述已自动生成

输入：

5,14

1,3,2,5,4

2,3,1,5,3

3,2,2,4

4,3,1,5,3

5,3,4,1,2

深度优先遍历序列：

1. 从1开始搜索
2. 搜索4
3. 搜索3
4. 搜搜2
5. 搜索5

*结果：1 4 3 2 5*

广度优先遍历序列：

1. 将1入队
2. 取出队尾元素1，1的邻接点2，4，5入队。现在队的情况：2，5，4
3. 取出队尾元素4，4的邻接点3入队。现在队的情况：3，2，5
4. 取出队尾元素5，5的邻接点都已经入队或访问过。现在队的情况：3，2
5. 取出队尾元素2，5的邻接点都已经入队或访问过。现在队的情况：3
6. 取出队尾元素3，3的邻接点都已经入队或访问过。现在队的情况：NULL

*结果：1 4 5 2 3*

* 使用graphviz做出如下图：

graph ethane {

1 -- 2 [type=s];

2 -- 3 [type=s];

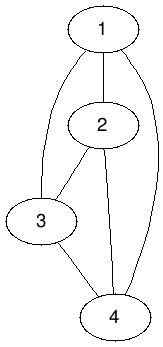
3 -- 1 [type=s];

3 -- 4 [type=s];

4 -- 1 [type=s];

2 -- 4 [type=s];

}



输入：

4,6

1,3,2,3,4

2,3,1,4,3

3,3,1,2,4

4,3,1,2,3

深度优先遍历序列：

1. 从1开始搜索
2. 搜索4
3. 搜索3
4. 搜搜2

*结果：1 4 3 2*

广度优先遍历序列：

1. 将1入队
2. 取出队尾元素1，1的邻接点2，3，4入队。现在队的情况：2，3，4
3. 取出队尾元素4，4的邻接点都已经入队或访问过。现在队的情况：2，3
4. 取出队尾元素3，3的邻接点都已经入队或访问过。现在队的情况：2
5. 取出队尾元素2，2的邻接点都已经入队或访问过。现在队的情况：NULL

*结果：1 4 3 2*

1. 实现

源代码：

<https://github.com/Reset12138/Projects_of_Data_Structures/tree/master/Experiment_5>

1. 总结

* 好像没啥总结的。。。太简单了