|  |
| --- |
| **寻找产生联系的最短距离路径** |
|  |
| **徐增武** |
|  |
|  |
|  |
|  |

摘要

这个文档介绍了如何寻找两个好莱坞明星联系的最短距离，这里我们使用宽度优先算法来寻找该最短距离，将在我们的程序里详细说明

引言

根据六度分割游戏可知，好莱坞电影业的任何人都可以通过六个步骤与凯文·培根联系起来，每个步骤都包括找到一部两位演员都主演的电影。

在这个问题中，我们感兴趣的是通过选择一系列连接两个演员的电影来找到两个演员之间的最短路径。例如，詹妮弗·劳伦斯和汤姆·汉克斯之间的最短路径是2：詹妮弗·劳伦斯和凯文·培根都主演了《X战警：头等舱》，凯文·培根和汤姆·汉克斯都主演了《阿波罗13》。

我们可以将其定义为一个搜索问题：我们的州是人民。我们的行为是电影，它把我们从一个演员带到另一个演员（一部电影确实可以把我们带到多个不同的演员面前，但这对这个问题没关系）。我们的初始状态和目标状态由我们试图联系的两个人定义。通过使用广度优先搜索，我们可以找到从一个参与者到另一个参与者的最短路径。

宽度优先搜索

概念

宽度优先搜索（又称广度优先搜索）是最简便的图的搜索算法之一，这一算法也是很多重要的图的算法的原型。Dijkstra单源最短路径算法和Prim最小生成树算法都采用了和宽度优先搜索类似的思想。其别名又叫BFS，属于一种盲目搜寻法，目的是系统地展开并检查图中的所有节点，以找寻结果。

* 1. 描述

宽度优先搜索和深度优先搜索对应，宽度优先搜索虽然也是遍历全局，但是遍历的方式是像波浪一样一层一层向外拓展的，而不是一头走到黑。走迷宫问题是BFS的一个典型应用。

* 1. 例子

如图1所示，比如我们从源节点1出发，要找到7的目的节点，应用宽度优先算法，我们先从1的所有展开节点进行寻找，如果没有，就继续从2的展开节点进行查找，以此类推，直到找到我们所需的目的节点。

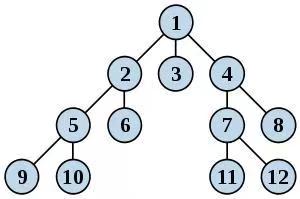


图1 拓扑图

队列

队列是一种特殊的线性表，特殊之处在于它只允许在表的前端进行删除操作，而在表的后端进行插入操作，和栈一样，队列是一种操作受限制的线性表。进行插入操作的端称为队尾，进行删除操作的端称为队头。队列中没有元素时，称为空队列。所以队列具有先进先出的特点。这也是我们后面算法所用到的原理。

1. 程序实现

## **载入数据**

将表格中的数据进行读取存储

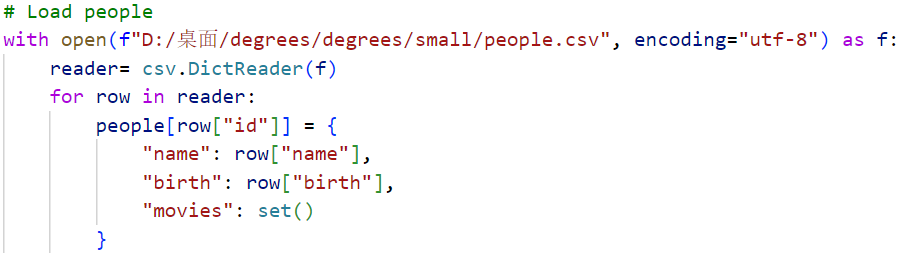


图2 载入数据函数

如图2所示，我们将表格里的数据进行读取存储，保存成字典格式，即一个键对应一个值。其他两个表格方法同上。

## **人名检索**

图3这个函数用于通过人名来检索任何人的id 和其他信息，并解决了同名情况的用户澄清问题。

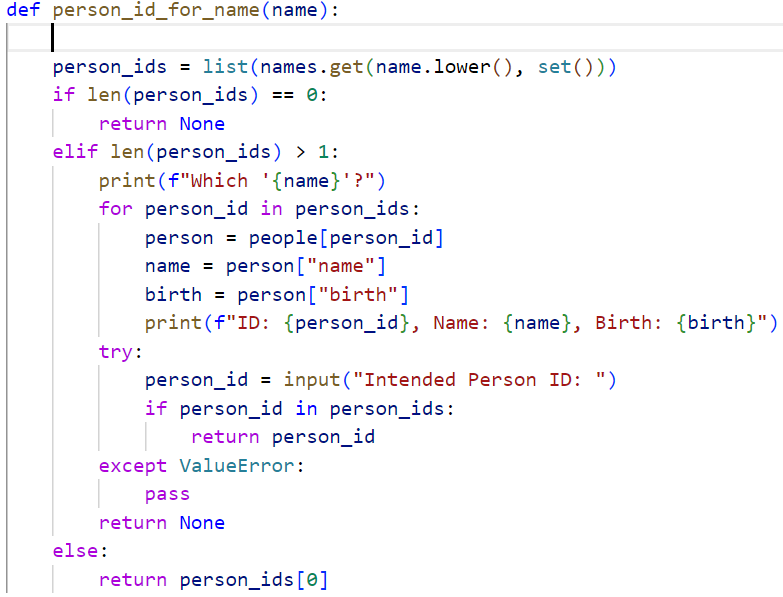


图3 检索函数

## **出演同一电影的人**

图4的函数用于寻找与指定人物出演过同一电影的人，并返回（电影id,人物id）。

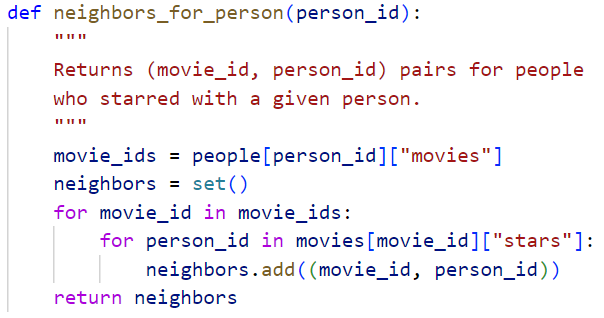


图4 寻找同一电影的人物

## **寻找两个人物的最短联系**

上面的函数部分都是源程序里已经给出的，本程序最关键的函数是需要我们自己填的，即寻找最短路径的函数。

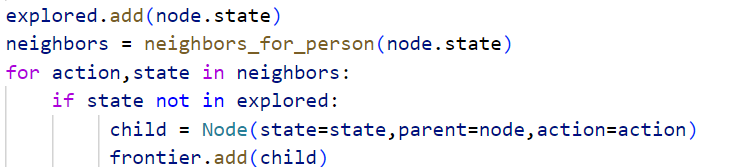


图5 队列遍历

如上图所示，我们将查看过的人物名添加到记录库，并将其子节点添加到队列里。直到找到目标节点。

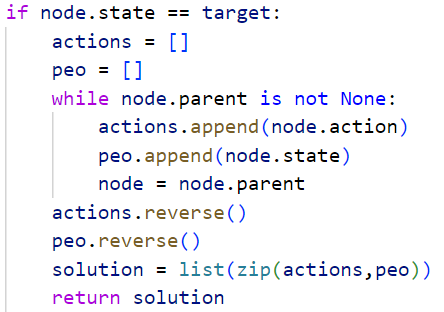


图6 返回路径输出值

当找到目标节点后，从目标节点往前推，依次返回其父节点和电影名的值。

致谢

感谢老师课堂上讲解内容的指导。因为这是我第一次去学习python，算是一个初学者，但是通过老师课堂上的讲解，让我对算法有了进一步了解，再通过课后自学python，虽然学的不是很深，但收获巨大，算是敲开了另一扇大门，也成功完成了本次的大作业。

参考文献

[1] Stuart J. Russell,Peter Norvig. 人工智能：一种现代方法.北京：人民邮电出版社,2010.10

[2] Python教程 ，2022年5月15号， https://www.liaoxuefeng.com/wiki/1016959663602400