

（深圳）

实验报告

开课学期： 大三上学期

课程名称： 人工智能

实验名称： 知识表示

实验性质： 设计

实验时间： 地点： 实验台号

学生专业：

学生学号： SZ170110202

学生姓名： 陈抒语

评阅教师：

报告成绩：

实验与创新实践教育中心印制

2019年10月

**小组贡献：25%**

**猴子摘香蕉：**

1. **简介/问题描述**
   1. **待解决问题的解释**

一个房间里，天花板上挂有一串香蕉，有一只猴子可在房间里任意活动（到处走动，推移箱子，攀登箱子等）。设房间里还有一只可被猴子移动的箱子，且猴子登上箱子时才能摘到香蕉，问猴子在某一状态下（设猴子位置为A，箱子位置为B，香蕉位置在C），如何行动可摘取到香蕉。

* 1. **问题的形式化描述**

将问题中的条件和要求转化成规则，由当前的状态按照规则不断转换成下一状态，直到完成题目要求的所达到的状态，找到完成任务的路径并将路径打印出来。

* 1. **解决方案介绍（原理）**

使用广度优先搜索，使用队列存储可达的所有状态，然后依次从队列中弹出，然后继续将新弹出的状态可按照规则到达的状态放入队列中，循环往复，直到可达状态为终止状态，然后按照顺序表path回溯打印状态转换过程，以及解决步骤。

1. **算法介绍**

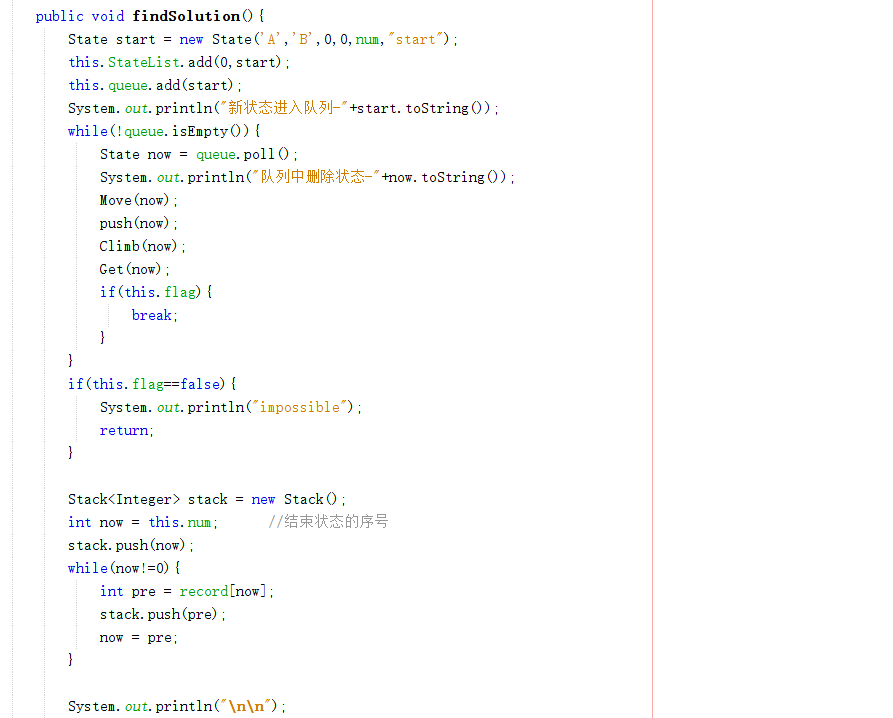
**2.1 所用方法的一般介绍**

宽度优先搜索算法（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。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。换句话说，它并不考虑结果的可能位置，彻底地搜索整张图，直到找到结果为止。

在图G=(V,E)上BFS的运行时间为扫描所有邻接表所花费时间至多为O(E)，初始化操作的开销为O(V)，因此过程BFS的全部运行时间为O(V+E)

使用顺序表path回溯，对每一个状态标上序号，并且在该状态所对应的record中写上到达该状态的上一状态的序号，然后通过堆栈，先进后出的性质，从结束状态的record开始压栈直到初始状态，然后在弹栈，打印搜索路径以及解决问题的方案。

**2.2 算法伪代码**



1. **算法实现**

**3.1 实验环境与问题规模**

实验环境：java jdk1.8

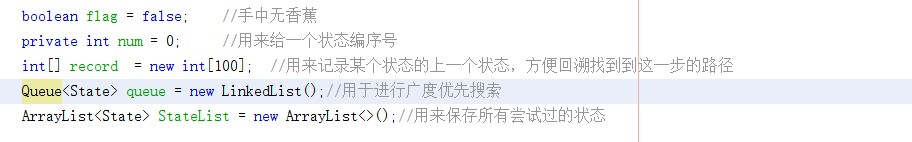
问题规模：BFS时间复杂度 O（V+E）

**3.2 数据结构**

模仿四元组存储当前状态的state class：



用于进行广度优先搜索以及回溯的数据结构：



**3.3 实验结果**

start

当前状态:State{monkey=A, box=B, on=0, getbanana=0}

Move from A to B

当前状态:State{monkey=B, box=B, on=0, getbanana=0}

pushBox fromBto C

当前状态:State{monkey=C, box=C, on=0, getbanana=0}

Climb

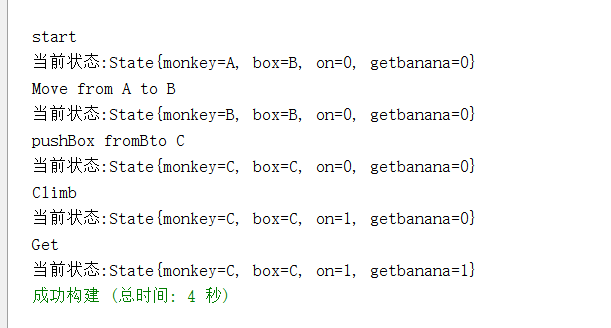
当前状态:State{monkey=C, box=C, on=1, getbanana=0}

Get

当前状态:State{monkey=C, box=C, on=1, getbanana=1}

**3.4 系统中间及最终输出结果（要求有屏幕显示）**

****



1. **总结及讨论（对该实验的总结以及任何该实验的启发）**

通过这次实验加深了对搜索算法理解，并且在搜索的过程中可以通过条件过滤掉一些状态，减少搜索的次数，锻炼的动手能力、编程能力，收获了关于搜索以及语义表达的方法，掌握了产生式以及使用元组来表达的发方法。并且发现，当初始状态较少的时候，使用深度优先搜索比广度优先搜索更加快速。