

**实验报告**

**题目：分油问题**

**学院：信息科学与技术学院**

**专业：计算机科学与技术**

**年级：2018级研究生交流生**

**课程：计算智能\_作业1**

**姓名： 余秋雨**

**学号：23020185755829**

**备注：**

**1.问题描述**

小孩分油问题：两个小孩去打油，一人带了一个。一斤的空瓶，另一个带了一个七两、一个三两的空瓶。原计划各打一斤油，可是由于所带的钱不够，只好两个人合打一斤油，可是由于所带的钱不够，只好两人合打一斤油，在回家的路上，两人平分一斤油，没有其他工具。试仅用三个瓶子（一斤、七两、三两）精确地分出两个半斤油。

将一斤油，用三个容量分别问一斤、七两、三两的瓶子，分出容量五两和五两的油，

**2.算法设计**

一、算法选择

结合深度优先、广度优先搜索算法的特点，选择广度优先算法来求解该分油问题。如果采用深度优先算法搜索，由于其盲目性可能导致搜索陷入局部陷阱，并不一定能求得解，即使得到解也不一定是最优解，因此不采用此算法。广度优先以牺牲空间代价和时间代价来换取保证取得最优解。由于该问题并不复杂，即使使用广度优先算法也不会占有太多的空间和时间，因此为了取得最优解这里选择广度优先算法来求解。

二、算法描述

（1）、建立一个队列，用来放入所有产生的新结点。其中记录每个状态三个瓶子里油量和每个结点的父亲。初始结点父亲设置为-1。

（2）建立一个链表，用来放入已产生的结点，用来与新状态比较，只需要比较前两个瓶子里的油量即可。

（3）开始操作，看目前形成的结点状态是不是要求的结果，是则停止搜索。如果不是，则按智能算子的规则求出新状态。

（4）判断新状态是否在搜索队列和查找链表中出现过，没有则加入该状态结点。

（5）队列头结点出队，返回步骤3。

三、合适的数据结构表示问题状态

（1）用向量（a、b、c）表示状态——a表示一斤瓶中的油量，b表示七两瓶中的油量，c表示三两瓶中的油量。

（2）问题的起始状态（10,0,0）

（3）问题的目标状态（5,0,0）

四、确定智能算子，用以变化状态的规则

|  |  |  |
| --- | --- | --- |
| 规则号 | 规则 | 解释 |
| 1 | (a,b) and a<7-> (7,b） | 七两瓶不满时装满 |
| 2 | (a,b) and b<3-> (a,3) | 三两瓶不满时装满 |
| 3 | (a,b) and a>0 ->(0,b) | 七两瓶不空时倒空 |
| 4 | (a,b) and b>0 ->(a,0) | 三两瓶不空是倒空 |
| 5 | (a,b) and a>0 and a+b<3 -> (0,a+b) | 七两瓶中的油全倒入三两瓶中 |
| 6 | (a,b) and b>0 and a+b<7 ->(a+b,0) | 三两瓶中的油全倒入七两瓶中 |
| 7 | (a,b) and a>0 and a+b>=3 ->(a+b-3,3) | 七两瓶中的油倒满三两瓶中 |
| 8 | (a,b) and c>0 and a+b>=7 ->(7,a+b-7) | 三两瓶中的油倒满七两瓶 |

**3.程序流程**

开始

定义初始状态

（10,0,0）

建立队列

建立链表

是否满足（5,5,0）？

否

结束

是

搜索

智能算子规则求新状态

是否是新状态？

是

否

**4.核心伪代码**

//倒入油的行为动作  
public void Add(int val)

{Current += val;

Remain -= val;

}  
//倒出油的行为动作  
public void Sub(int val)  
{Current -= val;

Remain += val;  
}

//广度优先搜索实现类  
public class WidthSearch  
public void S(TreeNode node)  
{while(unVisitedBttsArr.Count>=0) //未访问表中如果有结点继续循环搜索否则跳出  
{TreeNode n = (TreeNode)unVisitedBttsArr[0];  
bool flag = true;

//检查是否已经访问过  
for each(TreeNode i in haveVisitedBttsArr)

{if (i.Text.Equals(n.Text))  
{haveVisitedBttsArr.Add(unVisitedBttsArr[0]);  
unVisitedBttsArr.RemoveAt(0);  
flag = false;

break;

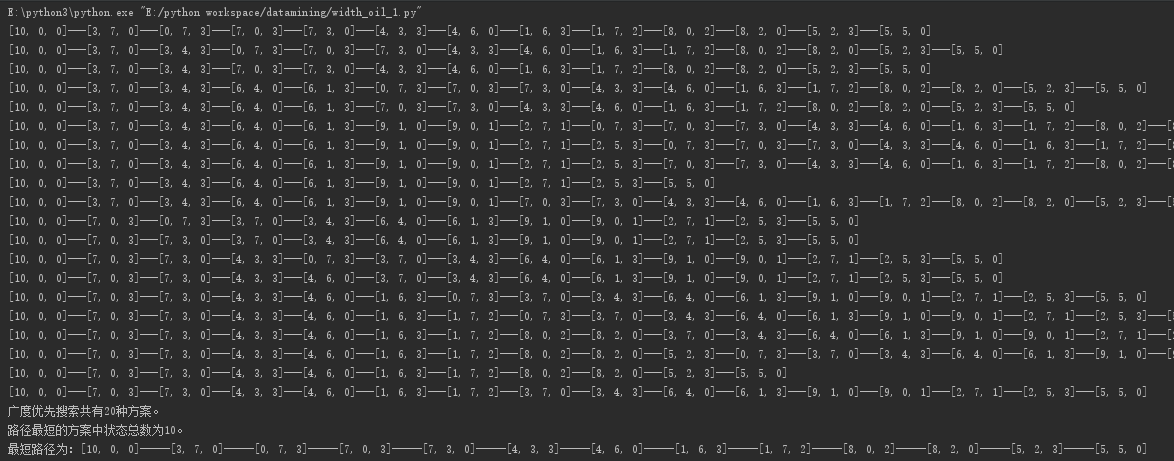
}

}  
//若已经遍历过的不需要继续遍历 跳到下一个  
if(flag)  
{if(Search(n))  
{  
return;}

}  
}  
}  
//创建子结点并加入到unVisitedBttsArr中。  
private bool CreateNode(TreeNode node)  
{TreeNode n = new TreeNode(BottlesState());  
unVisitedBttsArr.Add(n);  
if(bottle\_10.CurrentVal == 5)  
{node.Nodes.Add(n);  
SetPath(n);  
return true;  
}  
node.Nodes.Add(n);  
return false;  
}  
//回溯取得最佳路径  
private void SetPath(TreeNode n)  
{while(n.Parent!=null)  
{path = n.Text + " -> " + path;  
n.ForeColor = System.Drawing.Color.Blue;  
n = n.Parent;  
}

}

**5.代码运行及测试**



**6.结论**

深度优先搜索的搜索策略是扩展深度大的结点，直到找到目标结点。深度优先搜索过程中，为了避免死循环，从继续扩展的结点上删除下列结，不再存在后继的结点；状态相同的冗余或重复性结点；超过规定深度的结点。深度优先搜索的缺点是：未必能找到解，即使找到也未必是路径最短解。

广度优先搜索的搜索策略：总是在某一深度上先搜索所有结点，之后搜索下一个深度的结点。只要问题有解，广度优先搜索一定能在有限步内找到解且路径最短。广度优先搜索的弱点：随着深度的增加，结点数目指数增长，导致组合爆炸。

本次分油问题选择广度优先算法来求解该分油问题。如果采用深度优先算法搜索，由于其盲目性可能导致搜索陷入局部陷阱，并不一定能求得解，即使得到解也不一定是最优解，因此不采用此算法。广度优先以牺牲空间代价和时间代价来换取保证取得最优解。由于该问题并不复杂，即使使用广度优先算法也不会占有太多的空间和时间，因此为了取得最优解这里选择广度优先算法来求解。

采用python，建立列表，确定智能算子，采用广度优先搜索算法解决分油（5,5,0）。