# 实验1 约瑟夫（Josephu）问题

1. 约瑟夫问题

设编号为1，2，… m的m个人围坐一圈，从1开始报数，数到n 的那个人出列，它的下一位又从1开始报数，数到n的那个人又出列，依次类推，直到所有人出列为止，由此产生一个出队编号的序列。

2. 测试用例

(1) 总人数m,出列序号n。

(2) 测试数据

|  |  |  |
| --- | --- | --- |
| m | N | 出列顺序 |
| 8 | 6 | 64358721 |
| 7 | 5 | 5324716 |
| 6 | 1 | 123456 |
| 6 | 8 | 254163 |
| 1 | 1 | 1 |
| 1 | 0 | error |
| 4 | 2 | 2431 |
| 11 | 8 | 8532471110619 |
| 8 | 11 | 37562814 |
| 5 | 10 | 52314 |

3. 算法描述

（1）顺序存储算法

(a)将m 个人的位置用数组a的元素下标表示，数组元素的初始值为1,表示在圆圈内，出列为0。设当前出列人数total为0。

(b)从a的下标0位置开始递增报数，当报数达到n时，对a的该元素赋值为0，出列人数total加1。

(c)再从a的下一个下标位置开始，从1开始对a未出列元素继续报数，当报数达到n时，对a该位置的元素赋值为0，出列人数total加1。

(d)当出列人数total为m时，程序结束；否则，执行(c)。

**注： python 语言list列表是动态数组，可以删除元素。**

(2) 链表存储算法

（a）用循环链表存储m个人的位置信息。

（b）当报数到需要出列的结点时，删除该结点。

（c）当所有结点都出列后，循环结束。

4. 测试数据模拟算法执行过程

m=8 6

1 2 3 4 5 6 7 8

\*

第1 次出列 6 a[5]=0 total=1

1 2 3 4 5 7 8

\*

第2次出列4 a[3]=0 total=2

1 2 3 5 7 8

\*

第3次出列4 a[2]=0 total=3

1 2 5 7 8

\*

第4次出列5 a[4]=0 total=4

1 2 7 8

\*

第5次出列8 a[7]=0 total=5

1 2 7

\*

第6次出列7 a[6]=0 total=6

1. 2

\*

第7次出列2 a[1]=0 total=7

1

第8次出列1 a[0]=0 total=8

4. 实现：

代码我已放在GitHub仓库中，包含所有commit记录

<https://github.com/Reset12138/Projects_of_Data_Structures/tree/master/Experiment_1>

* 数组实现：（文件双击可打开）



电脑萤幕画面

描述已自动生成

* 链表实现：



电脑萤幕的截图

描述已自动生成

5. 扩展思考

1. 是否可以将出列顺序保存起来，需要的时候输出？

已直接实现。

1. 因为只关心最后出列的人员，是否还有更好的算法？

参考此博客[1]，我在不好理解的地方加入详细解释。

假设从1开始编号，则第一个出列的人的编号为

证明：

* 时。假设，。先数完一圈6个人后再数两个，即第人出列。
* 时。假设，。，即第2个人出列，符合实际。

首先我们把这m个人的序号编号从（由于n是可能大于m的，而当n大于等于m时，那么第一个出列的人编号是，而是可能等于0的，这样编号的话能够简化后续出列的过程），当数到的那个人出列，因此我们编号完成之后，开始分析出列的过程：  
第一次出列：  
一开始的时候，所有人的编号排成序列的模式即为：

那么第一次出列的人的编号则是。那么在第一个人出列之后，从他的下一个人又开始从0开始报数，为了方便我们设（为当前序列人数）那么在第一个人出列之后，则是下一次新的编号序列的首位元素，那么我们得到的新的编号序列为：（第一次已出列）

那么在这个新的序列中，第一个人依旧是从0开始报数，那么在这个新的序列中，每个人报的相应数字为：

那么第二次每个人报的相应数字与第一次时自己相应的编号对应起来的关系则为：

为当前序列的总人数，因为是循环的序列，可能大于总人数。  
那么这时我们要解决的问题就是个人的报数问题（即阶约瑟夫环的问题）

递归处理下去，直到最后只剩下一个人的时候，便可以直接得出结果。当我们得到一个人的时候（即一阶约瑟夫环问题）的结果，我们能根据这个结果，推导出二阶约瑟夫环的结果呢。即，当在解决m阶约瑟夫环问题时，序号为的人出列后，剩下的个人又重新组成了一个阶的约瑟夫环。

那么假如得到了这个阶约瑟夫环问题的结果为ans（即最后一个出列的人编号为ans），那么我们通过上述分析过程，可以知道，m阶约瑟夫环的结果：(m为当前序列的总人数)

而

联立可得m阶约瑟夫环的结果  
再进行化简。分类讨论：

* 时。那么上式则化简为：

即为：

而

得证：

* 时。易得上式可化简为：

即无论n和m大小关系如何，m阶约瑟夫环的结果都是：

这样我们就得到了递推公式。由于编号是从0开始的，那么我们可以令  
f[1] = 0； //当一个人的时候，出队人员编号为0  
f[m] = (f[m-1] + n)%m //n表示每次数到该数的人出列，m表示当前序列的总人数  
而我们只需要得到第m次出列的结果即可，那么不需要另外声明数组保存数据，只需要直接一个for循环求得m阶约瑟夫环问题的结果即可。由于往往现实生活中编号是从1-m，那么我们把最后的结果加1即可。

实现：



电脑萤幕画面

描述已自动生成

**参考文献**

1. 林夕-梦.约瑟夫环问题（最简单的数学解法）[EB/OL]. https://www.cnblogs.com/cmmdc/p/7216726.html, 2007–07–21/2020–05–30.