

实验 1 约瑟夫（Josephu）问题

王天一 320200931301 2022 年 4 月 10 日

注：该次实验的所有代码和文档均放入了压缩包以及我在 GitHub 的仓库：

[wtyqqq/Data_Structure_Lab \(github.com\)](https://github.com/wtyqqq/Data_Structure_Lab)

1 题目

设编号为 $1, 2, \dots, m$ 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

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，出列后数值为 0。设当前已出列人数 $total$ 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

常用方法：

(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) 当只有一个结点时，循环结束。

其它方法：

(1) 大循环内（小于 m 时一直循环），结点存储位置信息，单向链表遍历，元素出列，删除结点。

(2) 大循环内（小于 m 时一直循环），结点元素含位置信息，是否出列标志（圈内时为 1），单向链表遍历，出列元素对应结点的标志置为 0，出列元素个数加 1。

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 $a[0]=0$ $total=8$

5. 扩展思考

(1) 如果只关心最后出列的人员，不考虑数据结构要求，是否还有更快的算法？

有，使用递推公式法。

$$f(m, n) = (f(m-1, n) + n - 1) \bmod m + 1$$

$$f(m, n) = 0 \quad (m = 0)$$

使用此法能够大大减小算法的时间复杂度。

6. 代码实现

所有的代码实现在来 `lab1.py` 内，可供查阅。



lab1.py