

# 微视频

## 约瑟夫问题

中国海洋大学信息学院

魏振钢

Tel:0532-66781226

Email:wzgwzq@ouc.edu.cn

问题描述如下： $m$ 个人围成一圈，每个人手里有一个令牌（令牌值为一个正整数）从第一个人开始报数，数到 $n$ 的人出圈，同时将其令牌的值作为新的 $n$ 值；再由下一个人开始报数，数到 $n$ 的人出圈；.....依次输出出圈的人的编号。

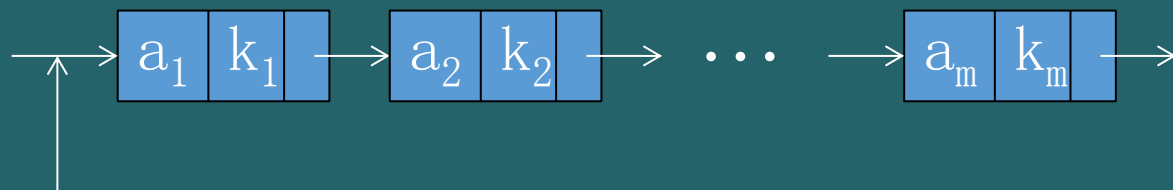
思考：



1. 如何表示这 $m$ 个人围成的圈？
2. 被数到的人出圈如何实现？
3. 算法如何实现？

1、可以考虑用不带头结点的循环单链表表示这一圈人。

设一个包括 $m$ 个结点的循环单链表，每个结点有三个数据域，分别保存：游戏者编号、令牌值、下一结点指针。



其中： $a_i$ 表示游戏者编号， $k_i$ 令牌值。

2、被数到的人出圈，就相当于在循环单链表中删除某个结点。

具体处理过程：

从第一个结点开始，依次遍历链表，并通过计数器计数，当计数器的值为 $n$ 时，输出该结点的所存编号（它即是应该出圈人的编号），并将该结点所存令牌值取出，设为新的 $n$ ，同时将计数器值归0。然后将该结点从链表中删除，使以后计数时不再起作用，相当于该人已出圈。再从下一个结点开始，依据上面的方法，如此继续，直到输出 $m$ 个值以后结束。



算法:

void Joseph(Node \*head,int n,int m)     //m:总人数 n:出列者报的令牌值, 初始值给定, head:指向最后一个结点的初始指针

{

int i,j,;

Node \*p,\*q;

q=head; p=q->next//p指向当前结点, 初值为首元结点, q为p的前驱

for (j=1;j<=m;j++)                     //逐个删除结点, 直至全部出列

{

for(i=1;i<n;i++)

{

q=p; p=p->next;                     //找到应出列的人

}

printf("% d",p->data);             //输出出列者序号

n=p->token;                     //n为下次应出人员令牌值

q->next=p->next; ;             //连接被删结点前后的两个节点

p=q->next;

free(q->next);                     //释放被删结点的存储空间

}

}