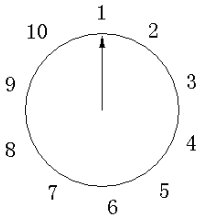
《Concrete mathematics 具体数学》第二版
1.3 The Josephus Problem 约瑟夫环问题
1.1.1版

翻译：刘炯(SharpMark)

2006年10月28日

我们最后一个例子是一个以Flavius Josephus命名的古老的问题[1]的变形，他是第一世纪一个著名的历史学家。据传说，如果没有Josephus的数学天赋，他就不可能活下来而成为著名的学者。在犹太—罗马战争中，他是被罗马人困在一个山洞中的41个犹太叛军之一，这些叛军宁死不屈，决定在罗马人俘虏他们之前自杀，他们站成一个圈，从一开始，依次杀掉编号是三的倍数的人，直到一个人也不剩。但是在这些叛军中的Josephus和他没有被告发的同伴觉得这么做毫无意义，所以他快速的计算出他和他的朋友应该站在这个恶毒的圆圈的哪个位置。[2]

在我们的变形了的问题中，我们以n个人开始，从1到n编号围成一个圈，我们每次消灭第二个人直到只剩下一个人。例如，这里我们以设$n = 10$做开始。



这时的消灭顺序是2, 4, 6, 8, 10, 3, 7, 1, 9。所以编号是5的人活下来了。这个问题：就是定位最后活下来的人的数字$J(n)$。

我们刚刚看到的是$J(10) = 5$的情况。我们可能会推测当n是偶数的时候$J(n) = n/2$；[3]而且当$n = 2$的时候结论验证了假设：$J(2) = 1$。但是其他一些数字比较小的情况告诉我们—当$n = 4$和$n = 6$的时候这个假设错误了。

n	1	2	3	4	5	6
$J(n)$	1	1	3	1	3	5

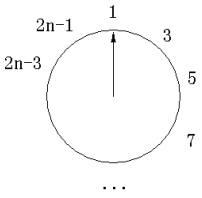
[1](Ahrens [5, vol.2] and Herstein 以及Kaplansky [187] 描述了这个故事的有趣历史，Josephus[197]自己对此事则有些漠然。)

[2]...因此也就保存下来了他的传奇，让我们可以听到。

[3]此处$n = 0$没有实际意义。

于是我们回到了起点[4]；让我们试着来个更好的猜测。呃...，$J(n)$看起来总是奇数。而且事实上，这个现象的原因是：围成的圆圈的第一轮消灭了所有的编号是偶数的人。之后我们又回到了与我们开始的时候类似的情形，除了人数只有原来一半的人，而且他们的编号改变了。

所以，让我们假设最开始有$2n$个人，在第一轮结束后，我们剩下



而且3将是下一个出局的。这个就像是以n个人开始的情况，除了每个人的编号变为两倍减一。那就是，[5]

$$J(2n) = 2J(n) - 1, \qquad \text{当 } n \geq 1.$$

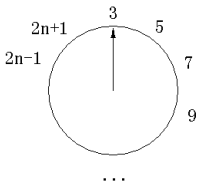
我们现在可以快速的计算一下当n比较大的时候的值。例如，我们知道$J(10) = 5$，所以

$$J(20) = 2J(10) - 1 = 2 \cdot 5 - 1 = 9.$$

同样可知$J(40) = 17$，进一步我们可以推算出

$$J(5 \cdot 2^m) = 2^{m+1} + 1.$$

但是，奇数的情况呢？[6]当有$2n + 1$个人的时候，编号是1的人会在第$2n$个人出局后紧接着出局，然后，我们剩下



我们再次获得了形如n个人的情况，但是这次他们的编号是两倍加一。所以

$$J(2n + 1) = 2J(n) + 1, \qquad \text{当 } n \geq 1.$$

与等式$J(1) = 1$组合，我们得到一个定义了J的所有取值的递推式:

$$\begin{aligned} J(1) &= 1; \\ J(2n) &= 2J(n) - 1, \qquad \text{当 } n \geq 1; \\ J(2n+1) &= 2J(n) + 1, \qquad \text{当 } n \geq 1; \end{aligned} \tag{1}$$

[4]即使如此，一个不好的猜想并不会浪费时间，因为它使我们深入问题。

[5]这里我们有更巧妙的格式：$J(2n) = newnumber(J(n))$，而$newnumber(k) = 2k - 1$.

[6]Odd case? Hey, leave my brother out of it.(不知何深意，故保留不译。—译者注)

这个公式不是从$J(n-1)$计算$J(n)$，这个递归式更加的“高效”，因为他以2为因子使n递减了一半或更多。我们可以计算$J(1,000,000)$，如上所示，我们只要应用19次(1)式。但是，我们依旧要寻找一个闭合形式[7]的公式，因为那样会更加快速和更有意义。总之，这是非常重要的事情。

用我们的递归式，我们可以很快地建造一张较小取值的表。可能我们可以通过这张表看出模式并猜出结果。

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$J(n)$	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15	1

瞧！看起来我们可以以2的幂分组(通过表中的竖线)；在每组的开始，$J(n)$总是1，而且在一组内以2递增。所以，如果我们把n写成形如$n=2^m+l$的公式，当2^m是不超过n的2的最大次幂，且l是剩余的数。这样我们的递归式的解法看起来是

$$J(2^m+l)=2l+1, \qquad \text{当 } m\geq 0\text{且}0\leq l<2^m. \tag{2}$$

(注意如果$2^m\geq n<2^{m+1}$，那么余下的数$l=n-2^m$满足不等式$0\leq l<2^{m+1}-2^m=2^m$。)

我们现在必须证明(2)式。如我们以前使用的数学归纳法一样，但是这次我们的数学归纳是基于变量m的。当$m=0$我们一定有$l=0$；因此(2)式的基础就变成了$J(1)=1$，这时是正确的。通过l是奇数还是偶数，这次的归纳证明分两部分。如果$m>0$且$2^m+l=2n$，那么l是偶数。且通过(1)式和归纳法假设可得

$$J(2^m+l)=2J(2^{m-1}+l/2)-1=2(2l/2+1)-1=2l+1,$$

这个就是我们想要的。奇数的情况证明方法类似，当$2^m+l=2n+1$。我们可能也注意到式子(1)还表达了这样一个关系

$$J(2n+1)-J(2n)=2.$$

总之，这个数学归纳法证明完毕，且(2)式被证明了。[8]

为了展示解法(2)式，让我们来计算一下$J(100)$。在这个例子中，我们有$100=2^6+36$，所以$J(100)=2\cdot 36+1=73$。

现在，我们解决了艰难的部分(解决问题)。我们再说点轻松的：每解决一个问题都可以泛化，使得可以应用一大类问题。当我们已经掌握一项技巧，完整的研究它，看看借助它我们可以走多远是非常值得的。所以，在这节的余下部分，我们将会研究我们的解法(2)式以及研究递归式(1)的一些扩展。这些研究将会展示出所有这样问题的结构和基础。

[7]递归式可以理解为开形式，非递归的可直接解出的式子为闭合形式。—译者注

[8]有个简单的方法！问题关键是对于任意m都有$j(2^m)=1$，且这个公式服从我们最开始的公式$J(2n)=2J(n)-1$。所以我们知道第一个人在人数是2的幂的时候一定会活下来。在一般的情况下，$n=2^m+l$，人数在处决了l个人之后减少到2的幂。这时剩下的第一个人就是最后活下来的人，是$2l+1$。

在我们寻找解法的时候，2的幂起到了重要的作用，所以很自然的，我们想用2的基数[9]表示n和$J(n)$。假设n的二进制表达式是

$$n = (b_m b_{m-1} ... b_1 b_0)_2;$$

也就是

$$n = b_m 2^m + b_{m-1} 2^{m-1} + \cdots + b_1 2 + b_0,$$

每个b_i取值是0或1，且最高位b_m是1。回想一下$n = 2^m + l$，我们先后得到如下表达式，

$$\begin{aligned} n &= (1 b_{m-1} b_{m-2} \cdots b_1 b_0)_2, \\ l &= (0 b_{m-1} b_{m-2} \cdots b_1 b_0)_2, \\ 2l &= (b_{m-1} b_{m-2} \cdots b_1 b_0 0)_2, \\ 2l+1 &= (b_{m-1} b_{m-2} \cdots b_1 b_0 1)_2, \\ J(n) &= (b_{m-1} b_{m-2} \cdots b_1 b_0 b_m)_2. \end{aligned}$$

(最后一个式子是因为$J(n) = 2l + 1$以及$b_m = 1$。)我们已经证明

$$J\Big((b_m b_{m-1} ... b_1 b_0)_2\Big) = (b_{m-1} ... b_1 b_0 b_m)_2; \tag{3}$$

这样，用计算机编程的专业术语解释就是我们从n计算$J(n)$只需要做一位循环左移！多么神奇。例如，如果$n = 100 = (1100100)_2$，那么$J(n) = J\Big((1100100)_2\Big) = (1001001)_2$，也就是$64 + 8 + 1 = 73$。如果我们从一开始就一直用二进制方法研究，我们可能会立即就发现这个模式。

如果我们以n个人开始，迭代[10]J函数$m + 1$次，计算机将会作$m + 1$次的一位循环左移；所以当n是一个$(m + 1)$位的数字，我们可能希望最后又得到n。但是实际上不是。举个例子，当$n = 13$，我们有$J\Big((1101)_2\Big) = (1011)_2$，但是之后$J\Big((1011)_2\Big) = (111)_2$，这时处理过程中断了；当0出现在首位的时候会被忽略。事实上，根据定义，$J(n)$必须总是$\leq n$，因为$J(n)$是存活着的人的编号；所以如果$J(n) < n$我们决不可能通过继续迭代回溯到n。

重复的应用过程J产生一个递减的序列值，最终到达一个“固定的值”，也就是当$J(n) = n$的时候。这种循环位移特点使得很容易观察出来这个固定的值将会是多少：迭代这个方法足够多的次数，总会产生每个位都是1的模式，其值为$2^{\nu(n)} - 1$，当$\nu(n)$是1-位在整个二进制序列的出现的次数。所以，当$\nu(13) = 3$，我们有

$$\overbrace{J(J(...J}^{\text{2或更多的}J}\ (13)...)) = 2^3 - 1 = 7;$$

[9] n的基数也就是n进制。—译者注

[10] (“迭代”这里的意思是一个函数使用其本身。)

同样的

$$\overbrace{J(J(...J}^{8\text{或更多}}(101101101101011)_2...)) = 2^{10} - 1 = 1023;$$

很奇怪，但是是正确的。[11]

让我们回到这个问题的最初猜想，也就是当n是偶数的时候$J(n) = n/2$。一般而言这个结论是明显但不正确的，但我们现在可以看看它到底在什么情况下是正确的：

$$\begin{aligned} J(n) &= n/2, \\ 2l+1 &= (2^m + l)/2, \\ l &= \frac{1}{3}(2^m - 2). \end{aligned}$$

如果这个式子$l = \frac{1}{3}(2^m - 2)$是一个整数，那么$n = 2^m + l$将会是一个解，因为l会小于2^m。不难验证当m是奇数，$2^m - 2$是3的倍数，但当m是偶数的时候不成立。(我们将会在第四章研究这个问题。)所以有无穷多个解可以满足$J(n) = n/2$，以如下形式开头：

m	l	$n = 2^m + l$	$J(n) = 2l + 1 = n/2$	n(二进制)
1	0	2	1	10
3	2	10	5	1010
5	10	42	21	101010
7	42	170	85	10101010

注意这个模式中最右边的一列。这些二进制数一位循环左移的结果和一位循环右移的结果一样。(结果都是减半)。

好了，我们对函数J了解的全面了；下一步是将这个问题一般化。如果我们的问题得到一个像(1)的递归式，但是常数不同，将会发生什么？我们可能就没有这么幸运能猜到答案了，因为答案可能很复杂。让我们通过引入常量α，β和γ[12]研究这个问题，并尝试为较一般的递归式找到一个闭合形式的公式

$$\begin{aligned} f(1) &= \alpha; \\ f(2n) &= 2f(n) + \beta, \qquad \text{当 } n \geq 1; \\ f(2n+1) &= 2f(n) + \gamma, \qquad \text{当 } n \geq 1. \end{aligned} \tag{4}$$

(我们开始的递归式中$\alpha = 1$，$\beta = -1$，且$\gamma = 1$。)以$f(1) = \alpha$开始计算，我

[11] Curiously enough, if M is a compact C^∞ n-manifold($n > 1$), there exists a differentiable immersion of M into $\mathbf{R}^{2n-\mu(n)}$ but not necessarily into $\mathbf{R}^{2n-\mu(n)-1}$. I wonder if Josephus was secretly a topologist?

[12] Looks like Greek to me.

们可以构建出来接下来的广义形式递归式的对n取值较小的表：

n	$f(n)$					
1	α					
2	2α	$+$	β			
3	2α			$+$	γ	
4	4α	$+$	3β			
5	4α	$+$	2β	$+$	γ	
6	4α	$+$	β	$+$	2γ	
7	4α			$+$	3γ	
8	8α	$+$	7β			
9	8α	$+$	6β	$+$	γ	

(5)

可以看出来α的系数是小于n的最大的2的幂。另外，在2的幂的范围内，β的系数每次减1，直到0。而γ的系数则从0开始每次增加1。所以，我们的表达式$f(n)$的形式是

$$f(n) = A(n)\alpha + B(n)\beta + C(n)\gamma, \tag{6}$$

通过与之相关的α，β和γ分离，可得

$$\begin{aligned} A(n) &= 2^m; \\ B(n) &= 2^m - 1 - l; \\ C(n) &= l. \end{aligned} \tag{7}$$

这里跟以往一样，$n = 2^m + l$且$0 \le l < 2^m$，当$n \ge 1$。

使用数学归纳法证明(6)式和(7)式并不是一件非常艰难的事情，但是这个计算过程是繁杂而且没有技术含量的。而且现在有个更好的方法来计算，通过带入选定具体的值，然后对比计算他们。[13]让我们通过具体的例子$\alpha = 1$，$\beta = \gamma = 0$来说明这个方法，当假设$f(n)$等于$A(n)$：递归式(4)变为

$$\begin{aligned} A(1) &= 1; \\ A(2n) &= 2A(n), \qquad \text{当 } n \ge 1; \\ A(2n+1) &= 2A(n) \qquad \text{当 } n \ge 1. \end{aligned}$$

可以肯定的是,$A(2^m + l) = 2^m$是正确的(通过对m进行数学归纳法可以证明)。

接下来，我们将反用递归式(4)以及解法(6)式。我们以一个简单的$f(n)$函数开始，看看是否存在常数(α, β, γ)可以确定它。将常函数$f(n) = 1$带入递归式(4)[14]可得

$$\begin{aligned} 1 &= \alpha; \\ 2 &= 2 \cdot 1 + \beta; \\ 3 &= 2 \cdot 1 + \gamma; \end{aligned}$$

[13]抓紧你的帽子，接下来的这部分是全新的内容。

[14]一个简洁的思想！

所以取值$(\alpha,\beta,\gamma)=(1,-1,-1)$满足这些等式，且有$A(n)-B(n)-C(n)=f(n)=1$。同样我们可以带入$f(n)=n$：

$$\begin{aligned} 1 &= \alpha; \\ 2n &= 2\cdot n+\beta; \\ 2n+1 &= 2\cdot n+\gamma; \end{aligned}$$

当$\alpha=1$，$\beta=2$且$\gamma=1$的时候，这些等式对于所有的n成立，所以我们不必用数学归纳法证明这些参数满足$f(n)=n$。我们已经*知道*$f(n)=n$在这个例子中是解，因为递归式(4)为每个n的取值唯一的定义了$f(n)$。

现在我们已经完成了核心的部分！我们已经表示了(6)式的函数$A(n)$，$B(n)$以及$C(n)$，他们解决了广义的(4)式，满足公式

$$\begin{aligned} A(n) &= 2^m, \qquad \text{当 } n=2^m+l \text{且} 0\le l<2^m; \\ A(n)-B(n)-C(n) &= 1; \\ A(n)+C(n) &= n. \end{aligned}$$

我们的对于(7)式的推测可以通过如下的式子立即解出来：
$C(n)=n-A(n)=l$以及$B(n)=A(n)-1-C(n)=2^m-1-l$。

这个过程已经接近于展示一个惊奇、有用的*repertoire method*来解决递归式问题。首先，我们要寻找我们的解法中已知的通用参数；这给予了我们能解得各个部分的特例。我们需要跟独立的参数数量(在这个例子中是三个，α，β和γ)同样多的独立的特解。练习16和20提供了更多的相关例子。[15]

我们知道原来的递归式J有一个神奇的解法，也就是二进制：

$$J\Big((b_m b_{m-1}...b_1 b_0)_2\Big)=(b_{m-1}...b_1 b_0 b_m)_2, \qquad \text{当 } b_m=1.$$

那么，广义的Josephus递归式也能如此神奇么？

当然，为什么不呢？我们可以把这个广义的递归式(4)重写成如下形式：

$$\begin{aligned} f(1) &= \alpha; \\ f(2n+j) &= 2f(n)+\beta_j, \qquad \text{当 } j=0,1 \text{且} n\ge 1, \end{aligned} \tag{8}$$

若我们设$\beta_0=\beta$且$\beta_1=\gamma$。这个递归式的展开式，二进制表示如下：

$$\begin{aligned} f\Big((b_m b_{m-1}...b_1 b_0)_2\Big) &= 2f\Big((b_m b_{m-1}...b_1)_2\Big)+\beta_{b_0} \\ &= 4f\Big((b_m b_{m-1}...b_2)_2\Big)+2\beta_{b_1}+\beta_{b_1} \\ &\ \ \vdots \\ &= 2^m f\Big((b_m)_2\Big)+2^{m-1}\beta_{b_{m-1}}+\ldots+2\beta_{b_1}+\beta_{b_0} \\ &= 2^m\alpha+2^{m-1}\beta_{b_{m-1}}+\ldots+2\beta_{b_1}+\beta_{b_0}. \end{aligned}$$

[15]注意：作者希望我们直观的了解"repertoire method"的例子，而不是给我们自顶向下的描述。这个方法最适合“线性”的递归式，这样解法就可以用n的一个由任意参数组合而成的和函数表示，例如等式(6)。等式(6)是关键。

假设我们现在不管[16]2的基数标记，随便让数字取代二进制中的0和1。这样就会得到

$$f\Big((b_m b_{m-1}...b_1 b_0)_2\Big) = (\alpha\beta_{b_{m-1}}\beta_{b_{m-2}}\dots\beta_{b_1}+\beta_{b_0})_2. \tag{9}$$

好了。如果我们把(5)式换一种写法[17]，我们会更早些看到这个模式：

n	$f(n)$
1	α
2	$2\alpha + \beta$
3	$2\alpha + \gamma$
4	$4\alpha + 2\beta + \beta$
5	$4\alpha + 2\beta + \gamma$
6	$4\alpha + 2\gamma + \beta$
7	$4\alpha + 2\gamma + \gamma$

举个例子，当$n = 100 = (1100100)_2$的时候，我们原来的Josephus取值为$\alpha = 1$，$\beta = -1$，且$\gamma = 1$带入后构造

n	$=$	$(1$	1	0	0	1	0	$0)_2$	$=$	100
$f(n)$	$=$	$(1$	1	-1	-1	1	-1	$-1)_2$		
	$=$	$+64$	$+32$	-16	-8	$+4$	-2	-1	$=$	73

结论跟之前一样。这个循环左移的特点之所以能满足是因为每个二进制数据块$(10...000)_2$在表示n的时候的时候转化为

$$(1\,{-1}...\,{-1}\,{-1})_2 = (0\,0\,...\,0\,1)_2.$$

所以我们改变曾给出的对递归式(8)的精简的解法(9)式。如果我们已经不受约束了，我们就可以使它更加一般化[18]，所以递归式

$$\begin{aligned} f(j) &= a_j, && \text{当} 1 \le j < d; \\ f(dn+j) &= cf(n)+\beta_j, && \text{当} 0 \le j < d \text{且} n \ge 1, \end{aligned} \tag{10}$$

如同之前的那个，除了我们的起始的数字的基数是d，产生值是基数c。所以，这个基数改变了的解法是

$$f\Big((b_m b_{m-1}...b_1 b_0)_d\Big) = (\alpha_{b_m}\beta_{b_{m-1}}\beta_{b_{m-2}}\dots\beta_{b_1}\beta_{b_0})_c. \tag{11}$$

[16]（“不管”=“删除”）

[17]I think I get it: The binary representations of $A(n)$, $B(n)$, and $C(n)$ have 1's in different positions.

[18]“有两种概括。一种是没有价值的，一种是有价值的。把一个小的问题套用一个大的术语是很简单的。从一些好的因素中提炼出精致的浓缩的部分更加困难。”—G. Pólya [297]

例如，假设一个幸运的巧合[19]下我们获得了一个递归式

$$\begin{aligned} f(1) &= 34, \\ f(2) &= 5, \\ f(3n) &= 10f(n)+76, \qquad \text{当} n \geq 1, \\ f(3n+1) &= 10f(n)-2, \qquad \text{当} n \geq 1, \\ f(3n+2) &= 10f(n)+8, \qquad \text{当} n \geq 1, \end{aligned}$$

假设我们想计算$f(19)$。这里我们有$d = 3$和$c = 10$。现在$19 = (201)_3$，通过这个基数改变了的解法，我们把数字一位一位的替换，把每个数字从基于基数3的表示方法替换为基数是10的。所以，使2变为5，并假设0和1变成76和-2，代入

$$f(19) = f\Big((201)_3\Big) = (576\ {-2})_{10} = 1258,$$

这就是我们的结果。

所以Josephus以及犹太—罗马战争留给我们一些有趣的广义递归式。[20]

译者的话：

第一次翻译数学文章，第一次使用LaTeX编辑文章。选择了《具体数学》的第1.3节翻译。因为这节看望让我眼前一亮。希望他也能让对数学感兴趣的你眼前一亮。翻译成中文是希望能有更多的人看到这篇文章。边学习LaTeX边翻译，花了两天的时间。不过翻译得不好，有什么错误的地方还希望可以指出来。

对于内容，我看了很多遍，每一遍都有新的感觉和更深的体会。不过仍然不敢说完全掌握了，如果你在阅读本文或者原书的时候有什么想法、问题欢迎跟我交流。TeX源文件可以到我的个人主页下载。

http://www.sharpmark.net/

sharpmark@gmail.com

最后说一下关于注释，如果加了“译者注”字样的，就是我翻译的，如果没有加的，就是原书的旁批。据说这些旁批是上这门课的作者的学生们写的课堂笔记，呵呵。

版本更新：

1.1.1 [07-12-23] 更新了个人的新的联系方式。

1.1.0 [06-11-08] 又重新通读了全文，修正了一些错误。

1.0.1 [06-10-30] 修正了公式(9)上面那个公式中的一个字符错误。

刘炯

于大连理工大学

2006年10月28日

[19]可能这是一个坏的运气.

[20]但是实际上，我反对战争的递归。