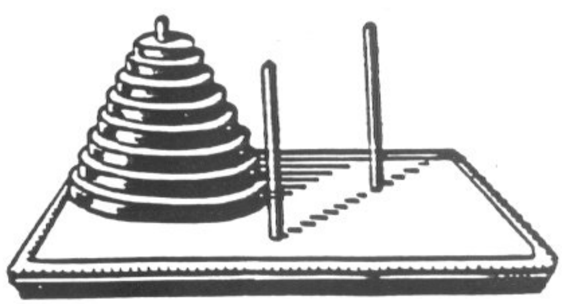
Contents

We began by covering the three problems presented in the first chapter of Concrete Mathematics ([CM]) by Graham, Knuth, and Patashnik: the Tower of Hanoi, a problem concerning lines in the plane, and the Josephus problem.

1 The Tower of Hanoi

The Tower of Hanoi puzzle is attributed to the French mathematician Edouard Lucas, who came up with it in 1883. His formulation involved three pegs and eight distinctly-sized disks stacked on one of the pegs from the biggest on the bottom to the smallest on the top, like so:



The goal is to move the stack of disks one disk at a time to one of the other pegs, with the restriction that a larger disk cannot be on top of any smaller disks at any point, in as few moves as possible.

Here we will consider the general case, i.e. supposing we’ve got n disks, all stacked in decreasing order from bottom to top on one of three pegs, what’s the minimum number of (legal) moves needed to move the stack to one of the other pegs?

We’ll start by expressing the minimum number of moves required to move a stack of n disks as a recurrence relation (i.e. in terms of the minimum number of moves required to move a stack with less than

n disks). Then, we'll find a closed-form expression for the minimum number of moves required, and prove that the closed-form and recurrent expressions are equivalent.

1.1 Recurrent Solution

Note that in order to move the bottom disk, we need to move all the disks above it to another peg first. Then we can move the bottom disk to the remaining empty peg, and move the $n-1$ smaller disks back on top of it. In other words, one way to move the stack to one of the two empty pegs is to first move the top $n-1$ pegs to the unchosen peg, move the bottom disk to the chosen peg, and then move the top $n-1$ pegs to the chosen peg. To move the top $n-1$ pegs to the chosen peg, we can simply undo the steps we used to move it to the unchosen peg, where the "initial peg" is the chosen peg instead of the peg where the tower originally was. Hence, letting T_n denote the minimum number of moves required to move a tower with n disks to a given peg, we know that

$$T_n \leq 2T_{n-1} + 1, \text{ where } n \geq 1. \tag{1}$$

This is so because, since we know that a tower with n disks can be moved to a chosen peg by

1. moving the top $n-1$ disks to the unchosen peg (in T_{n-1} moves),
2. moving the bottom disk to the chosen peg (1 move), and
3. finally, moving the top $n-1$ pegs from the unchosen to the chosen peg (another T_{n-1} moves, since we can simply reverse what we did to get them to the unchosen peg),

the minimum number of moves T_n required to move the n disks to a chosen peg is at most $2T_{n-1}+1$ (it may be less). In other words, $2T_{n-1}+1$ moves are sufficient to move the stack of n disks.

We now claim, however, that $2T_{n-1}+1$ moves are also necessary to move the stack of n disks, i.e. that

$$T_n \geq 2T_{n-1} + 1, \text{ where } n \geq 1. \tag{2}$$

One can argue as follows. In order to move the largest (bottom) disk anywhere, we have to first get the $n-1$ smaller disks on top of it onto *one* of the other pegs (since the bottom disk is the biggest, it cannot be moved onto a nonempty peg). This will take at least T_{n-1} moves. Once this is done, we have to move the bottom disk at least once; we may move it more than once. After we're done moving the bottom disk, we have to move the $n-1$ other disks back on top of it, which will take at least T_{n-1} moves, in order to complete the transfer of the stack to the chosen peg. From this, it follows that we need at least $2T_{n-1}+1$ moves to transfer the stack to the chosen peg, i.e. that $2T_{n-1}+1$ moves are necessary to move the stack of n disks. Hence from (1) and (2), along with the boundary condition $T_0 = 0$ (i.e. it takes at least no moves to move a stack with no disks to another peg), we get the following recurrent solution for the minimum number of moves T_n required to move a tower with n disks to another peg:

$$T_n = \begin{cases} 0\,, & \text{if } n = 0; \\ 2T_{n-1} + 1\,, & \text{if } n > 0. \end{cases} \tag{3}$$

1.2 Closed-Form Solution

An annoying thing about the recurrent solution (3) is that, to calculate T_n, we need to calculate T_k for $k = 1, \ldots, n-1$ first. It's easy to see that for large n, this can be *really* annoying. What we'd like instead is a solution $f(n)$ into which we can simply plug n to get the minimum number of moves required. One way to get such a solution is to first come up with a guess, and then prove that the guess is in fact a solution. In particular, given a guess $f(n)$ we'd want to prove that $f(n) = T_n$ for all $n \in \mathbb{N} \cup \{0\}$, where T_n is defined by (3). Since this is a statement about the nonnegative integers, we can prove it using mathematical induction.

One way to guess a solution is to write out a couple of the terms and see if any pattern thereby makes itself apparent. We have

$$T_0 = 0,\ T_1 = 1,\ T_2 = 3,\ T_3 = 7,\ T_4 = 15,\ T_5 = 31,\ T_6 = 63, \ldots$$

Based on this, one may conjecture that

$$T_n = 2^n - 1 \ \forall\, n \in \mathbb{N} \cup \{0\}.$$

We now prove that this is in fact the solution.

Proof. We proceed via induction on n.

Base. Let $n = 0$. Then by definition, $T_0 = 0$. Also, $2^0 - 1 = 1 - 1 = 0$. Hence the conjecture holds for $n = 0$.

Inductive Step. Suppose that the conjecture is true for all $n < k$, where $k \in \mathbb{N} \cup \{0\}$. Then

$$\begin{aligned} T_k &= 2T_{k-1} + 1 \\ &= 2(2^{k-1} - 1) + 1 \\ &= 2^k - 2 + 1 \\ &= 2^k - 1. \end{aligned}$$

Hence by the axiom of mathematical induction, $T_n = 2^n - 1 \ \forall\, n \in \mathbb{N} \cup \{0\}$. □

Finally, we note an interesting way to find a closed-form solution to the Tower of Hanoi recurrence (3) without having to guess that the solution is $T_n = 2^n - 1$. Consider what happens when we add 1 to T_n:

$$T_n + 1 = \begin{cases} 1\,, & \text{if } n = 0; \\ 2T_{n-1} + 2 = 2(T_{n-1} + 1)\,, & \text{if } n > 0. \end{cases}$$

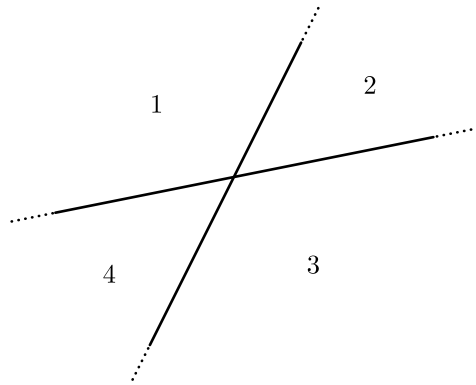
Now, letting $U_n = T_n + 1$, we get the following recurrence:

$$U_n = \begin{cases} 1\,, & \text{if } n = 0; \\ 2U_{n-1}\,, & \text{if } n > 0. \end{cases}$$

It's pretty easy (in any case easier than for T_n) to see that the solution to this recurrence is $U_n = 2^n$. Since $U_n = T_n + 1$, we get $T_n = U_n - 1 = 2^n - 1$.

2 Lines in the Plane

The second problem we will consider is as follows: what's the maximum number of regions L_n that can be defined in the plane by n lines? For $n = 0$, it's easy to see that there's only one region ($L_0 = 1$), and for $n = 1$ there're two regions no matter how the line's oriented ($L_1 = 2$). If $n = 2$, then the maximum number of regions we can define is $L_2 = 4$:



Four regions is the best we can do with two lines because the lines must either cross or not cross; if they cross, then the lines define four regions, and if they don't cross they define three.

2.1 Recurrent Solution

Since we have $L_0 = 1$, $L_1 = 2$, and $L_2 = 4$, one might be led to conjecture that $L_n = 2^n$. This immediately breaks down when we consider $n = 3$, however; no matter how the third line's placed, we can only split at most three pre-existing regions, i.e. we can add at most three new regions using the third line.

This can be generalized as follows. Suppose that $n - 1$ lines have already been drawn. First of all, note that adding a new line adds k new regions *if and only if* the new line crosses k of the old regions. Also, the new line crosses k of the old regions *if and only if* it hits the old lines in $k - 1$ different places[1]. Now, since two lines can intersect in at most one point, the new line can hit the $n - 1$ old lines in at most $n - 1$ distinct points. Combined with the two preceding equivalences, this means that adding a new line can add at most n regions, i.e.

$$L_n \leq L_{n-1} + n, \quad \text{for } n > 0.$$

Actually, we also have

$$L_n \geq L_{n-1} + n, \quad \text{for } n > 0.$$

One can argue as follows. First, suppose $n = 1$. Then the inequality holds (trivially), since $L_1 = 1 = 0 + 1 = L_0 + 1$. Next, suppose we've already drawn $n - 1$ lines in a way that defines L_{n-1} regions. Note that if we were to draw the n^{th} line such that it's parallel to one of the old lines, then we'd miss out on intersecting that line; hence draw the n^{th} line such that it's not parallel to any of the $n - 1$ old lines. Also, make sure that the new line doesn't intersect two old lines at the same point, i.e. it doesn't hit any intersection points between the old lines. A new line placed in this way then hits $n - 1$ old lines in $n - 1$ distinct points, which means that the new line has added n new regions to L_{n-1}, i.e. $L_n \geq L_{n-1} + n$.

Hence we have the following recurrent solution to the problem:

$$L_n = \begin{cases} 1 & \text{if } n = 0; \\ L_{n-1} + n & \text{if } n > 0. \end{cases} \tag{4}$$

[1] If the new line crosses k old regions, then since each of the old regions is bounded by an old line, the new line must have hit $k - 1$ boundaries, i.e. $k - 1$ old lines. Conversely, if the new line hits $k - 1$ of the old lines, then pick a direction along the new line and start from "infinitely far away" and proceed towards the first hit encountered in that direction. Each time the new line crosses an old line, the new line crosses into a new region. Hence after $k - 1$ hits the new line has crossed over from the first old region into $k - 1$ other old regions, i.e. the total number of regions the new line lies in is $1 + k - 1 = k$.

2.2 Closed-Form Solution

This time, it's harder to see a general pattern based on the first few terms:

$$L_0 = 1,\ L_1 = 2,\ L_2 = 4,\ L_3 = 7,\ L_4 = 11,\ L_5 = 16, \ldots$$

So, let's try "unfolding" the recurrent solution (4) instead:

$$\begin{aligned}
L_n &= L_{n-1} + n \\
&= L_{n-2} + (n-1) + n \\
&= L_{n-3} + (n-2) + (n-1) + n \\
&\ \vdots \\
&= L_0 + 1 + 2 + \cdots + (n-2) + (n-1) + n \\
&= 1 + \sum_{i=1}^{n} i \\
&= 1 + \frac{n(n+1)}{2}.
\end{aligned}$$

We now prove that L_n does in fact equal $1 + \frac{n(n+1)}{2}$ for all $n \in \mathbb{N} \cup \{0\}$:

Proof. We proceed via induction on n.

Base. Let $n = 0$. Then $L_0 = 1$ by definition, and $1 + \frac{0(0+1)}{2} = 1$.

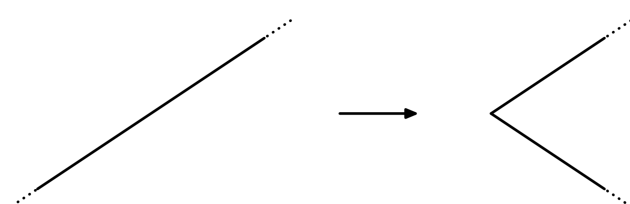
Inductive Step. Suppose that $L_k = 1 + \frac{k(k+1)}{2}$ for all $k = 1, 2, \ldots, n-1$. Then

$$\begin{aligned}
L_n &= L_{n-1} + n \\
&= 1 + \frac{(n-1)n}{2} + n \\
&= 1 + \frac{1}{2}n^2 - \frac{1}{2}n + n \\
&= 1 + \frac{1}{2}n^2 + \frac{1}{2}n \\
&= 1 + \frac{n(n+1)}{2}.
\end{aligned}$$

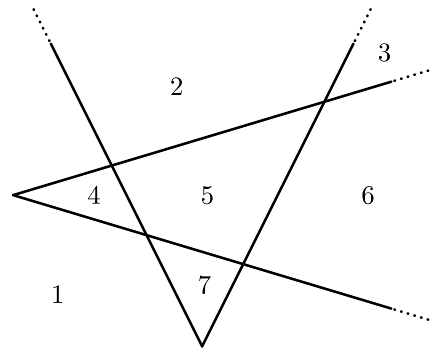
Hence by the axiom of mathematical induction, $L_n = 1 + \frac{n(n+1)}{2}\ \forall\, n \in \mathbb{N} \cup \{0\}$. □

2.3 Using Lines with a Single Bend

Before moving on, let's consider a slight variation of the original problem: what if instead of using lines, we use lines with a *single* "bend" in them (in the following, a "bent line" means a line with a single bend):



What's the maximum number of regions Z_n in the plane that can be defined with n bent lines? Intuitively, we can get more regions with fewer lines, because the bend can capture extra regions; for example, $Z_2 = 7$:



A key observation here is that a single bent line is like two intersecting straight lines, except that the parts of the lines on one side of their intersection have been "chopped off". Hence, for example, the maximum number of regions that can be defined using a single bent line is equal to $L_2 - 2$, where L_2 is the maximum number of regions that can be defined using two straight lines. It turns out (see pg. 8 and Exercise 18 in Chapter 1 of [CM]) that the recurrent solution is

$$Z_n = L_{2n} - 2n, \quad \text{for } n \geq 0.$$

Using the closed-form solution we got for L_n, the closed-form solution to this recurrence is

$$\begin{aligned} Z_n &= 1 + \frac{2n(2n+1)}{2} - 2n \\ &= 2n^2 - n + 1, \quad \text{for } n \geq 0. \end{aligned}$$

Hence for large n, since the dominating term in L_n is $\frac{1}{2}n^2$ and the dominating term in Z_n is $2n^2$, we can get about $\frac{2}{1/2} = 4$ times as many regions using bent lines compared to using straight lines.

3 The Josephus Problem

The Josephus problem is named for the first century historian Flavius Josephus. Josephus was allegedly trapped in a cave by Romans, along with 41 other Jewish Rebels, during the Jewish-Roman war. The rebels concluded that they preferred suicide to being captured by the Romans, and decided to form a circle and have every third person kill himself. Josephus, however, quickly calculated where he and a friend should stand in the circle so as to not have to commit suicide, and hence lived to tell the tale.

The "Josephus problem" considered here is actually a variant of Josephus's problem; in particular, given n people numbered from 1 to n standing in a circle, if we go around the circle eliminating every *second* person until only one person is left, what's the survivor's number $J(n)$?

3.1 Recurrent Solution

First, suppose that the number of people n in the circle is even, i.e. $n = 2\ell$ for some $\ell \in \mathbb{N}$. Note that all of the even-numbered people will be eliminated first:

$$1,\ \mathbf{2},\ 3,\ \mathbf{4}, \ldots,\ \boldsymbol{2\ell - 2},\ 2\ell - 1, \boldsymbol{2\ell}$$

After person 2ℓ is eliminated, we're left with the ℓ odd-numbered people $1, 3, \ldots, 2\ell - 3, 2\ell - 1$. The key observation here is that the position of the surviving person out of these ℓ people is given by $J(\ell)$. Since the number of the k^{th} person around the circle with the ℓ odd-numbered people is $2k-1$, the number of the surviving person out of the 2ℓ people is simply $2J(\ell)-1$.

Next, suppose that the number of people n in the circle is odd, i.e. $n = 2\ell + 1$ for some $\ell \in \mathbb{N}$. Again, all of the even-numbered people will be eliminated first:

$$1,\ \mathbf{2},\ 3,\ \mathbf{4},\ldots,\ \mathbf{2\boldsymbol{\ell}-2},\ 2\ell-1, \mathbf{2\boldsymbol{\ell}}, 2\ell+1$$

However, this time person 1 (instead of person 3) will be the first odd-numbered person eliminated. After eliminating person 1, we're left with the ℓ people $3, 5, \ldots, 2\ell-1, 2\ell+1$. Again, the position of the surviving person out of these ℓ people is given by $J(\ell)$, and since the k^{th} person's number in this group of ℓ people is given by $2k+1$, the number of the surviving person out of the $2\ell+1$ people is simply $2J(\ell)+1$.

Letting $J(1) = 1$, the recurrent solution is therefore

$$J(n) = \begin{cases} 1 & \text{if } n = 1; \\ 2J(\ell) - 1 & \text{if } n = 2\ell,\ \text{where } \ell \in \mathbb{N}; \\ 2J(\ell) + 1 & \text{if } n = 2\ell + 1,\ \text{where } \ell \in \mathbb{N}. \end{cases} \tag{5}$$

3.2 Closed-Form Solution

Here it's productive to first list some terms, grouping them as follows:

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

This suggests the following guess: if we write $n = 2^m + r$, where 2^m is the largest power of two not exceeding n and r is the remainder, then

$$J(n) = J(2^m + r) = 2r + 1,\ \text{where } m \geq 0 \text{ and } 0 \leq r < 2^m. \tag{6}$$

The restriction on r comes from the fact that, if $2^m \leq n < 2^{m+1}$, then since $n = 2^m + r \implies r = n - 2^m$, we get $0 \leq r < 2^{m+1} - 2^m = 2^m(2-1) = 2^m$.

Of course, we need to prove that (6) is equivalent to (5):

Proof. We proceed via induction on m.

Base. Suppose $m = 0$. Then $r = 0$, and $2(0) + 1 = 1 = J(1)$.

Inductive Step. Suppose that, for all n such that $n = 2^k + r$, where $k \in \{1, 2, \ldots, m-1\}$, we have $J(n) = J(2^k + r) = 2r + 1$. We then have two cases:

(1) Suppose $m > 0$ and $n = 2^m + r = 2\ell$, where $\ell \in \mathbb{N}$ (i.e. n is even). Then $r = 2\ell - 2^m = 2(\ell - 2^{m-1})$. Since $\ell - 2^{m-1} \in \mathbb{Z}$, this means r is even, and so

$$\begin{aligned} J(2^m + r) &= J\left(2\underbrace{\left(2^{m-1} + \frac{r}{2}\right)}_{\ell}\right) \\ &= 2J\left(2^{m-1} + \frac{r}{2}\right) - 1 && \text{(by (5))} \\ &= 2\left[2\left(\frac{r}{2}\right) + 1\right] - 1 && \text{(by the inductive hypothesis, since } 0 \leq r/2 < 2^{m-1}\text{)} \\ &= 2r + 2 - 1 \\ &= 2r + 1. \end{aligned}$$

(2) Suppose $m > 0$ and $n = 2^m + r = 2\ell + 1$, where $\ell \in \mathbb{N}$ (i.e. n is odd). Then $r = 2\ell - 2^m + 1 = 2(\ell - 2^{m-1}) + 1$, which means that r is odd. Hence, since

$$\begin{aligned} n &= 2^m + r \\ &= 2\left(2^{m-1} + \frac{r}{2}\right) \\ &= 2\left(2^{m-1} + \frac{r-1}{2}\right) + 1, \end{aligned}$$

and $2^{m-1} + \frac{r-1}{2} \in \mathbb{N} \cup \{0\}$ (since r is odd), we have

$$\begin{aligned} J(2^m + r) &= 2J\left(2^{m-1} + \frac{r-1}{2}\right) + 1 && \text{(by (5))} \\ &= 2\left[2\left(\frac{r-1}{2}\right) + 1\right] + 1 && \text{(by the inductive hypothesis, since } 0 \leq (r-1)/2 < r/2 < 2^{m-1}) \\ &= 2(r - 1 + 1) + 1 \\ &= 2r + 1. \end{aligned}$$

Hence by the axiom of mathematical induction, $J(n) = 2r + 1$, where $n = 2^m + r$, $m \geq 0$, and $0 \leq r < 2^m$. □

3.3 Closed-Form Solution in Radix 2

The radix 2 (binary) representation of the number n of people in the circle is

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

which we'll denote by

$$n = (b_m b_{m-1} \cdots b_1 b_0)_2,$$

where $b_i \in \{0, 1\}$ for $i = 0, 1, \ldots, m$. But since $n = 2^m + r$, where 2^m is the largest power of 2 not greater than n, we have

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

i.e.

$$n = (1 b_{m-1} b_{m-2} \cdots b_1 b_0)_2$$

and

$$r = (0 b_{m-1} b_{m-2} \cdots b_1 b_0)_2.$$

This implies that

$$2r = (b_{m-1} b_{m-2} \cdots b_1 b_0 0)_2,$$

and, using the closed-form solution (6) for $J(n)$ that we found in the previous section, that

$$J(n) = 2r + 1 = (b_{m-1} b_{m-2} \cdots b_1 b_0 1)_2.$$

Finally, since $b_m = 1$, we have

$$J(n) = J((b_m b_{m-1} \cdots b_1 b_0)_2) = ((b_{m-1} b_{m-2} \cdots b_1 b_0 b_m)_2.$$

Hence to get the number of the person out of n people that will survive, all we have to do is write n in binary and perform a one-bit cyclic shift to the left.

3.3.1 Iterated Application of the J Function

Now let's consider what happens when we iteratively apply the J function. We might initially expect that after $m+1$ applications of J, we get n again, since each application of J involves a one-bit cyclic shift to the left. This doesn't work, however; in particular, since by definition $J(n) \leq n$, we have $J(J(n)) \leq J(n) \leq n$, $J(J(J(n))) \leq J(J(n)) \leq J(n) \leq n$, etc. This means that if at any point $J(k) < k \leq n$, we can never recover n by continuing to apply the J function iteratively.

So, what's actually happening when we iterate the J function? It turns out that whenever a 0 becomes the leading bit, that bit is simply dropped. For example,

$$J(11) = 7 \implies J((1011)_2) = (111)_2.$$

In particular,

$$J(7) = 7 \implies J((111)_2) = (111)_2.$$

In general, after a certain number of iterative applications of J, we'll reach a "fixed point" where the continued application of J will always return the same value. In particular, this fixed point can be found by simply dropping all of the 0's in the binary representation of n. Doing so gives us the binary number $(11\cdots 1)_2$ containing $\nu(n)$ 1's, whose value is $2^{\nu(n)} - 1$.[2]

3.3.2 When is $J(n)$ Equal to $n/2$?

In [CM], the first instance of this problem that was examined was when $n = 10$. Seeing that $J(10) = 5$, one may have been led to conjecture that $J(n) = \frac{n}{2}$ if n is even. While this is certainly not true in general (e.g. $J(8) = 1$), we'll now consider when this is actually true.

Recalling that $n = 2^m + r$ and the closed-form solution (6) for $J(n)$, we have

$$\begin{aligned} J(n) = \frac{n}{2} &\iff 2r + 1 = \frac{2^m + r}{2} \\ &\iff 4r + 2 = 2^m + r \\ &\iff 3r = 2^m - 2 \\ &\iff r = \frac{2^m - 2}{3}. \end{aligned}$$

Since $\frac{2^m-2}{3} < 2^m$, this means that if $\frac{2^m-2}{3} \in \mathbb{N}$ as well, then $n = 2^m + \frac{2^m-2}{3}$ will be such that $J(n) = \frac{n}{2}$. Clearly, $2^m - 2$ cannot be divisible by 3 if m is even (since this would imply that $2^m - 2$ is even); one can prove (see [CM] Chapter 4) that if m is odd, then $2^m - 2$ is divisible by 3. Hence whenever m is odd, if we let $r = \frac{2^m-2}{3}$, then we'll have $J(n) = \frac{n}{2}$. The table below gives n and $J(n)$ for $m = 1, 3, 5, 7$:

m	r	$n = 2^m + r$	$J(n) = 2r + 1 = \frac{n}{2}$	n (binary)
1	0	2	1	10
3	2	10	5	1010
5	10	42	21	101010
7	42	170	85	10101010

In particular, note that the n for which $J(n) = \frac{n}{2}$ are those for which a one-bit cyclic shift to the left is equivalent to halving n ("ordinary-shifting one place right").

[2] Note that $x = (11\cdots 1)_2$ ($\nu(n)$ 1's) is equal to $2^{\nu(n)-1} + 2^{\nu(n)-2} + \cdots + 2 + 1$. This means that $2x = 2^{\nu(n)} + 2^{\nu(n)-1} + \cdots + 2^2 + 2$. Hence $2x - x = x = 2^{\nu(n)} - 1$.

3.4 Generalized Josephus Recurrence

We were able to successfully guess the correct closed-form solution to the recurrence (5). In general, however, it might not be as easy to guess correctly. In this section we'll illustrate the *repertoire method* for solving recurrences by considering the general recurrence of the form

$$f(n) = \begin{cases} \alpha, & \text{if } n = 1; \\ 2f(\ell) + \beta, & \text{if } n = 2\ell, \text{ where } \ell \in \mathbb{N}; \\ 2f(\ell) + \gamma, & \text{if } n = 2\ell + 1, \text{ where } \ell \in \mathbb{N}. \end{cases} \tag{7}$$

The values of $f(n)$ for a few n are given in the table below:

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		

Writing

$$f(n) = A(n)\alpha + B(n)\beta + C(n)\gamma, \tag{8}$$

and writing $n = 2^m + r$ as before, it seems that

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

Hence we can conjecture that a closed-form solution to (7) is

$$f(n) = f(2^m + r) = 2^m\alpha + (2^m - 1 - r)\beta + r\gamma. \tag{9}$$

3.4.1 Repertoire Method for Solving Recurrences

Of course, we can prove that (9) is in fact a solution to (7) using induction, but since that'll turn out to be messy, we'll instead proceed by plugging particular constants α, β, and γ, or a particular function f, into the original recurrence (7) and finding the corresponding function or constants, respectively, that make the plugged-in thing(s) work with (7). After plugging in constants/a particular $f(n)$ three times, we'll get three sets of constants & functions and, using (8), we'll get three equations in the three unknowns $A(n)$, $B(n)$, and $C(n)$. Solving this system will in turn give us a closed-form solution to the original recurrence (7).

First, let's plug $\alpha = 1$ and $\beta = \gamma = 0$ into (7). This gives us the recurrence

$$f(n) = \begin{cases} 1, & \text{if } n = 1; \\ 2f(\ell), & \text{if } n = 2\ell, \text{ where } \ell \in \mathbb{N}; \\ 2f(\ell), & \text{if } n = 2\ell + 1, \text{ where } \ell \in \mathbb{N}. \end{cases}$$

Using (8), we also have

$$f(n) = A(n).$$

Hence we can rewrite the above recurrence as

$$A(n) = \begin{cases} 1, & \text{if } n = 1; \\ 2A(\ell), & \text{if } n = 2\ell, \text{ where } \ell \in \mathbb{N}; \\ 2A(\ell), & \text{if } n = 2\ell + 1, \text{ where } \ell \in \mathbb{N}. \end{cases}$$

The first few terms are

n	ℓ	$A(n)$
1	0	1
2	1	2
3	1	2
4	2	4
5	2	4
6	3	4
7	3	4
8	4	8
9	4	8

It seem as if

$$A(n) = A(2^m + r) = 2^m. \tag{10}$$

Here's a proof that this is correct:

Proof. We proceed by induction on m.

Base. Suppose $m = 0$. Then $n = 1$, and $2^0 = 1 = A(1)$.

Inductive Step. Suppose that for all $k \in \mathbb{N}$ such that $k \leq m - 1$, we have $A(2^k + r) = 2^k$.

(1) Suppose $n = 2^m + r$ is even. Then r is even as well, and

$$\begin{aligned} A(2^m + r) &= A\left(2\left(2^{m-1} + \frac{r}{2}\right)\right) \\ &= 2A\left(2^{m-1} + \frac{r}{2}\right) \\ &= 2 \cdot 2^{m-1} \\ &= 2^m. \end{aligned}$$

(2) Suppose $n = 2^m + r$ is odd. Then r is odd as well, and

$$\begin{aligned} A(2^m + r) &= A\left(2\left(2^{m-1} + \frac{r}{2}\right)\right) \\ &= A\left(2\left(2^{m-1} + \frac{r-1}{2}\right) + 1\right) \\ &= 2A\left(2^{m-1} + \frac{r-1}{2}\right) \\ &= 2 \cdot 2^{m-1} \\ &= 2^m. \end{aligned}$$

Hence by the axiom of mathematical induction, $A(n) = A(2^m + r) = 2^m$ for all $n \in \mathbb{N}$. □

Next, let's plug $f(n) = 1$ into (7). This gives us

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

We can see by inspection that this implies $\alpha = 1$ and $\beta = \gamma = -1$. Hence from (8), we have

$$1 = A(n) - B(n) - C(n). \tag{11}$$

Finally, plugging $f(n) = n$ into (7), we get

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

This means, again by inspection, that $\alpha = 1, \beta = 0$, and $\gamma = 1$. By (8), we then have

$$n = A(n) + C(n). \tag{12}$$

In summary, we've obtained the system of equations

$$\begin{aligned} A(n) &= 2^m, \quad \text{where } n = 2^m + r \text{ and } 0 \le r < 2^m; && (10) \\ A(n) - B(n) - C(n) &= 1; && (11) \\ A(n) + C(n) &= n. && (12) \end{aligned}$$

Plugging (10) into (12), we get

$$\begin{aligned} C(n) &= n - 2^m \\ &= 2^m + r - 2^m \\ &= r. \end{aligned}$$

Finally, plugging $C(n) = r$ and (10) into (11) gives us

$$B(n) = 2^m - 1 - r.$$

Hence, as we had conjectured, a general solution to the recurrence (7) is

$$f(n) = f(2^m + r) = 2^m\alpha + (2^m - 1 - r)\beta + r\gamma.$$

3.4.2 Using "Relaxed" Radix 2 Notation

Recall the generalized Josephus Recurrence (7):

$$f(n) = \begin{cases} \alpha, & \text{if } n = 1; \\ 2f(\ell) + \beta, & \text{if } n = 2\ell, \text{ where } \ell \in \mathbb{N}; \\ 2f(\ell) + \gamma, & \text{if } n = 2\ell + 1, \text{ where } \ell \in \mathbb{N}. \end{cases}$$

Letting $\beta_0 = \beta$ and $\beta_1 = \gamma$, we can rewrite this more compactly as

$$f(n) = \begin{cases} \alpha, & \text{if } n = 1; \\ 2f(\ell) + \beta_j, & \text{where } n = 2\ell + j, \ j \in \{0, 1\}. \end{cases}$$

In particular, j is the value of the rightmost bit in the radix 2 representation of n. Recalling that $n = 2^m + r$, we have

$$\begin{aligned} n &= 2^m + r \\ &= 2^m + b_{m-1}2^{m-1} + \cdots + 2b_1 + b_0 \\ &= 2(2^{m-1} + b_{m-1}2^{m-2} + \cdots + 2b_2 + b_1) + b_0 \\ &= 2\,\underbrace{(1b_{m-1}\ldots b_2b_1)_2}_{\ell} + \underbrace{b_0}_{j}. \end{aligned}$$

Hence

$$\begin{aligned} \boxed{f(n)} &= f((1b_{m-1}\ldots b_1b_0)_2) \\ &= 2f((1b_{m-1}\ldots b_2b_1)_2) + \beta_{b_0} \\ &= 2[2f(1b_{m-1}\ldots b_3b_2)_2 + \beta_{b_1}] + \beta_{b_0} \\ &\ \ \vdots \\ &= 2^m f(1) + 2^{m-1}\beta_{b_{m-1}} + 2^{m-2}\beta_{b_{m-2}} + \cdots + 2\beta_{b_1} + \beta_{b_0} \\ &= \boxed{2^m\alpha + 2^{m-1}\beta_{b_{m-1}} + 2^{m-2}\beta_{b_{m-2}} + \cdots + 2\beta_{b_1} + \beta_{b_0}.} \end{aligned}$$

We can write this result succinctly in "binary" notation as

$$\boxed{f(n) = (\alpha\beta_{b_{m-1}}\beta_{b_{m-2}}\ldots\beta_{b_1}\beta b_0)_2.} \tag{13}$$

Recall the table of values of the generalized Josephus recurrence that we generated earlier:

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

We can check our new result (13) by using it to recreate this table of values (where $\beta_0 = \beta$ and $\beta_1 = \gamma$:

n	$f(n)$						
$1 = (1)_2$							$2^0\alpha$
$2 = (10)_2$					$2^1\alpha$	$+$	$2^0\beta$
$3 = (11)_2$					$2^1\alpha$	$+$	$2^0\gamma$
$4 = (100)_2$			$2^2\alpha$	$+$	$2^1\beta$	$+$	$2^0\beta$
$5 = (101)_2$			$2^2\alpha$	$+$	$2^1\beta$	$+$	$2^0\gamma$
$6 = (110)_2$			$2^2\alpha$	$+$	$2^1\gamma$	$+$	$2^0\beta$
$7 = (111)_2$			$2^2\alpha$	$+$	$2^1\gamma$	$+$	$2^0\gamma$
$8 = (1000)_2$	$2^3\alpha$	$+$	$2^2\beta$	$+$	$2^1\beta$	$+$	$2^0\beta$
$9 = (1001)_2$	$2^3\alpha$	$+$	$2^2\beta$	$+$	$2^1\beta$	$+$	$2^0\gamma$

Example 1. Let $n = 100 = (1100100)_2$. Then, letting $\alpha = 1, \beta_0 = \beta = -1$, and $\beta_1 = \gamma = 1$ (i.e. using the original Josephus recurrence), we have

$$\begin{aligned}
f(100) &= f((1100100)_2) \\
&= 2^6(1) + 2^5(1) + 2^4(-1) + 2^3(-1) + 2^2(1) + 2^1(-1) + 2^0(-1) \\
&= 64 + 32 - 16 - 8 + 4 - 2 - 1 \\
&= 73 \\
&= J(100),
\end{aligned}$$

as expected. ∎

Finally, we note that the cyclic-shift property of the J function noted earlier follows from the substitution of $\alpha = 1, \beta_0 = -1$, and $\beta_1 = \gamma = 1$ into (13). First, applying f with these coefficients to n is equivalent to taking the radix 2 representation of n and changing every 0 to a -1. Now, it turns out that this is equivalent to turning every block of binary digits $(10\ldots0)_2$ in the radix 2 representation of n into $(0\ldots01)_2$, which is equivalent to performing a one-bit cycle shift left on n. Here's a proof that $(1{-1}{-1}\ldots{-1})_2 = (00\ldots01)_2$:

Proof. Suppose $n = (1{-1}{-1}\ldots{-1})_2$. Then

$$\begin{aligned}
n &= 2^m - 2^{m-1} - 2^{m-2} - \cdots - 2 - 1 \\
&= 2^{m-1}(2-1) - 2^{m-2} - \cdots - 2 - 1 \\
&= 2^{m-1} - 2^{m-2} - \cdots - 2 - 1 \\
&= 2^{m-2}(2-1) - 2^{m-3} - \cdots - 2 - 1 \\
&= 2^{m-2} - 2^{m-3} - \cdots - 2 - 1 \\
&\;\;\vdots \\
&= 2 - 1 \\
&= 1 \\
&= (00\ldots01)_2.
\end{aligned}$$

□

Example 2. Again letting $\alpha = 1, \beta_0 = \beta = -1$, and $\beta_1 = \gamma = 1$, we have

$$\begin{aligned} f(100) &= f((1100100)_2) \\ &= (11{-}1{-}11{-}1{-}1)_2 \\ &= (1001001)_2 \\ &= 73 \\ &= J(100). \end{aligned}$$

■