Homework 1, Problem 1

1. Problem Description: Write details of pages 12-13, the discussion of cyclic properties of $J(n)$ and the false guess that $J(n) = \frac{n}{2}$. $J(n)$ is the index of survivor in the Josephus Problem.
2. Already Known: The recurrence in the Problem is:
   $J(1) = 1$;
   $J(2n) = 2J(n) - 1$, for $n \geq 1$
   $J(2n+1) = 2J(n) + 1$, for $n \geq 1$
   and the solution to the recurrence is:
   $J(2^m + l) = 2l + 1$, for $m \geq 0$ and $0 \leq l < 2^m$
3. Goal: Explore some generalizations of the recurrence in Josephus Problem and uncover the structure that underlies all such problems.

Introducing the Binary Representation

1. The table of small input values:

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

2. It is obvious that the powers of 2 played and important role in our solution, so let's look at the binary representations of n and $J(n)$.
3. Suppose n's binary representation is $n = (b_m b_{m-1} ... b_1 b_0)_2$ which is equal to the equation,
   $n = b_m 2^m + b_{m-1} 2^{m-1} + ... + b_1 2^1 + b_0$
   Here, all the b_i's are either 0 or 1 and the bit b_m must be 1.

Compute $J(n)$ from n under Binary Representation(1)

1. According to the solution of the recurrence, we denote n with the equation:
   $n = 2^m + l$,
   therefore, we have:
   $n = (1b_{m-1}...b_1b_0)_2$,
   note that the difference between the expressions of n lies in the leading bit, where b_m is changed to 1.
2. $l = n - 2^m$, therefore, we can change the 1 in the leading bit of the binary representation of n to 0 and denote l as follows:
   $l = (0b_{m-1}b_{m-1}...b_1b_0)_2$
3. Now let's consider the right hand side of the solution equation $J(2^m + l) = 2l + 1$, obviously, we need to denote $2l$.

Compute $J(n)$ from n under Binary Representation(2)

1. Now let's see how to denote $2l$. In the binary representation, to multiply a number by 2, we only need to do a **left shift 1 bit** operation, therefore, we can denote $2l$ as follows:
   $2l = (b_{m-1}b_{m-1}...b_1b_00)_2$
2. It is obvious to get the representation of $2l + 1$:
   $2l + 1 = (b_{m-1}b_{m-1}...b_1b_01)_2$
3. The only digit that is in
   $n = (b_mb_{m-1}...b_1b_0)_2$
   but not in $2l + 1$ is b_m. Also the only digit that is in $2l + 1$ but not in n is 1. But notice that $b_m = 1$!!! Therefore,
   $2l + 1 = (b_{m-1}b_{m-1}...b_1b_0b_m)_2$

Compute $J(n)$ from n under Binary Representation(3)

1. The relation between n, m and l is:
   $J(n) = J(2^m + l) = 2l + 1$, for $m \geq 0$ and $0 \leq l < 2^m$.
2. Here is the binary representation of n and $J(n) = 2l + 1$
   $n = (b_m b_{m-1} ... b_1 b_0)_2$
   and,
   $2l + 1 = J(n) = (b_{m-1} b_{m-1} ... b_1 b_0 b_m)_2$
3. We proved that
   $J((b_m b_{m-1} ... b_1 b_0)_2) = (b_{m-1} b_{m-1} ... b_1 b_0 b_m)_2$
4. Conclusion: We can get $J(n)$ from n by **doing a 1-bit cyclic shift left**.
5. Example: if $n = 200 = (11001000)_2$, then
   $J(n) = J((10010001)_2) = 128 + 16 + 1 = 145$, which means that the people with index 145 will be the only survivor.

Why we can not expect to end up with n again?

1. If we start with n and apply the J function to n itself for $(m+1)$ times, since n is an $(m+1)$-bit bumber, we may expect to end up with n again. i.e. $J(J(...J(n)...)) = n$
2. This is actually impossible, since when 0 is shifted to the leading bit, it disappears.
3. Example: $n = 100 = (1100100)_2$, we have $J((1100100)_2) = (1001001)_2$, but then $J((1001001)_2) = (0010011)_2 = (10011)_2$ where the process breaks down.
4. Actually $J(n)$ must always be $\leq n$ by definition, since $J(n)$ is the survivor's number, hence the only case that we can get back up to n by continuing to iterate is when $J(n) = n$. If $J(n) < n$, we are not able to do that.

The Result of the Iteration of J

1. Repeated application of J produces a sequence of decreasing values that eventually reach a fixed point, which is a pattern of all 1's with a value of $2^k - 1$, where k is the number of 1 bits in the binary representation of n.
2. Why? Since during the iteration, the 0's will be continuously thrown away when they are at the leading bit until all the 0's are deleted. What remains is the string composed of 1's, where no matter how many times we do the 1-bit cyclic shifting, the value will not change any more. This value is the fixed point.
3. Example:
   $J(J(\ldots J(100)\ldots)) = J(J(\ldots J((1100100)_2)\ldots)) = 2^3 - 1 = 7$
   Here, the k for 100 $k_{100} = 3$.
   $J(J(\ldots J((101101101101011)_2)\ldots)) = 2^{10} - 1 = 1023$
   Here, the k for $(101101101101011)_2$ is 10.

When Is Our Guess $J(n) = \frac{n}{2}$ *true?(1)*

1. The Guess: $J(n) = \frac{n}{2}$ when n is even.
2. From the table, we know that the guess is obviously false. But, when $n = 2$ and $n = 10$, the guess is true.

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

3. Now, we are interested in determining exactly when it is true. And we already know these equations:
   $J(n) = J(2^m + l) = 2l + 1$, $J(n) = \frac{n}{2}$ (Our Guess)
   and, $n = 2^m + l$.

When Is Our Guess $J(n) = \frac{n}{2}$ *true?(2)*

1. From the 3 equations we have, it is easy to get:
   $J(n) = 2l + 1 = \frac{n}{2} = \frac{(2^m + l)}{2} \Longrightarrow l = \frac{1}{3}(2^m - 2)$
2. All right, here we get to know that l must be less than 2^m and greater than or equal to 0 when $m > 0$. When $m = 0$, it is obvious that $\frac{n}{2}$ is not an integer. So we ignore this case.
3. Now, the only requirement for our guess $J(n) = \frac{n}{2}$ to be true is that l should be an integer. For a specific m, if l is an integer by the equation $l = \frac{1}{3}(2^m - 2)$, then our guess is true.
4. So what we need to do now is to find out when $(2^m - 2)$ is a multiple of 3.

when is $(2^m - 2)$ *a multiple of 3?(1)*

1. First of all, look at the table below in which there is the value of $(2^m - 2)$ for the first 9 integers.

m	1	2	3	4	5	6	7	8	9
$(2^m - 2)$	0^*	2	6^*	14	30^*	62	126^*	254	510^*

   Note that all the $(2^m - 2)$'s with a $*$ can be divided exactly by 3. The corresponding m's are 1, 3, 5, 7 and 9.

2. Conjecture: $(2^m - 2)$ is a multiple of 3 when m is odd, but not when m is even.

when is $(2^m - 2)$ *a multiple of 3?(2)*

1. Prove the conjecture by mathematical induction. In the base case, when $m = 1$, 0 is a multiple of 3; when $m = 2$, 2 is not a multiple of 3. Therefore, the base case is true.
2. Assume that when $m = k$, the conjecture is true: If k is odd, $(2^k - 2)$ is a multiple of 3; otherwise, $(2^k - 2)$ is not.
3. Now, we prove the case when $m = k + 2$ (not $m = k + 1$ since we need to keep the parity of the induction variable k).
   $(2^{(k+2)} - 2) = 4 \times 2^k - 2 = 4 \times (2^k - 2) + 6$.
   We know that 6 is a multiple of 3, so we only need to care about whether $4 \times (2^k - 2)$ is a multiple of 3 or not. We can prove that it is true directly from our induction assumption. Since 4 is not a muptiple of 3, we only consider $(2^k - 2)$. When k and $k + 2$ are both odd, $(2^k - 2)$ is a multiple of 3. Otherwise not.
4. Till now, we proved that our conjecture is true.

When Is Our Guess $J(n) = \frac{n}{2}$ true?(3)

1. When m is odd, $(2^m - 2)$ is a multiple of 3. When m is even, $(2^m - 2)$ is a not a multiple of 3.
2. For each m which is odd, we have a corresponding l which makes our guess $J(n) = \frac{n}{2}$ true. Therefore, there are infinitely many solutions to the equation $J(n) = \frac{n}{2}$.
3. Here in the table are some examples:

m	l	$n = 2^m + 1$	$J(n) = 2l + 1 = \frac{n}{2}$	n(binary)
1	0	2	1	10
3	2	10	5	1010
5	10	42	21	101010
7	42	170	85	10101010
9	170	682	341	1010101010

When Is Our Guess $J(n) = \frac{n}{2}$ *true?(4)*

1. In the table, notice the pattern in the rightmost column. These are the binary numbers for which cyclic shifting produces the same result as ordinary shifting one place right.
2. Example: $(101010)_2 \Rightarrow_{leftcyclicshifting} (10101)_2$ and $(101010)_2 \Rightarrow_{ordinaryshifting1digitright} (10101)_2$
3. Left Cyclic Shifting: Compute the $J(n)$ value for an given n.
4. Ordinary 1 Digit Right Shifting: Compute the value of $\frac{n}{2}$ when the last digit of n is "0".
5. Our guess is $J(n) = \frac{n}{2}$, so we need to make sure that the Left Cyclic Shifting produces the same result as the Ordinary 1 Digit Right Shifting.
6. We can conclude that when $n = (1010...1010)_2$, our guess $J(n) = \frac{n}{2}$ is true.

Thank you!

Shang Yang
syang@cs.sunysb.edu
02/14/2008