Problem for 2000 March

Proposed by Dan Jurca

This two-part problem involves $b_n$, the number of binary trees with $n$ nodes ($0 \leq n$), so may interest students of computer science. Recall

\[
\begin{align*}
b_0 &= 1 \\
b_1 &= 1 \\
b_2 &= 2 \\
b_3 &= 5,
\end{align*}
\]

and more generally one may compute $b_n$ recursively as follows

\[
b_0 = 1, \quad 1 \leq n \Rightarrow b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i},
\]

or using the closed formula

\[
0 \leq n \Rightarrow b_n = \frac{1}{n+1} \binom{2n}{n}.
\]

a. Show that $b_n$ equals an odd integer if and only if there exists a nonnegative integer $p$ such that $n = 2^p - 1$.

b. The table printed below shows the time required to calculate $b_n$ for the first few values of $n$ using a C program, also shown below, which evaluates $b_n$ recursively. Examination of the table shows that if $3 \leq n$, then the time to compute $b_n$ closely approximates 3 times the time to compute $b_{n-1}$. Explain this.

(The proposer compiled this program with the GNU compiler, and ran it under Red Hat Linux version 6.1 on a computer with a 200 MHz Pentium Pro microprocessor. Here ``usec'' means ``microsecond''.)

Solution by the proposer
a. We shall prove the assertion by induction on \( n \), observing that it hold for \( n \leq 5 \). Suppose that \( 6 \leq n \) and that \( 0 \leq m < n \Rightarrow b_m \) equals an odd integer if and only if there exists \( q \) such that \( m=2^q-1 \). Assume first that \( b_n \) equals an odd integer. Using the inductive hypothesis we have

\[
 b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i} \\
\equiv b_0 b_{n-1} + b_1 b_{n-2} + b_3 b_{n-4} + b_7 b_{n-8} + \ldots + b_{n-2} b_1 + b_{n-1} b_0 \pmod{2}.
\]

Now \( b_i \equiv 1 \pmod{2} \) if and only if \( i=2^j-1 \), and \( b_{n-1-i} \equiv 1 \pmod{2} \) if and only if \( n-1-i=2^k-1 \). Hence each term in the sum above equals 0 (mod 2) unless \( n=2^j+2^k-1 \) for some \( j \) and \( k \). Since, by assumption, \( b_n \equiv 1 \pmod{2} \), there do exist \( j \) and \( k \) such that \( n=2^j+2^k-1 \). Now if \( j \neq k \), then the summation has two equal terms which therefore cancel mod 2, so it follows that \( j=k \). Hence \( n=2^j+2^j-1 = 2^{j+1}-1 \), so that the assertion holds also for \( n \).

Next, if \( n=2^p-1, 2 \leq p \), then since \( b_n=\sum_{i=0}^{n-1} b_i b_{2^p-2-i} \), the only terms which contribute an odd value to the sum are those for which \( i=2^j-1 \), some \( j \), and \( 2^p-2-i=2^p-2^j-1=2^k-1 \), some \( k \); i.e., \( 2^p=2^j+2^k \). Then \( j=k=p-1 \), so the only term which contributes an odd value to the sum is the one with \( i=2^p-1 \), so that \( b_n \) equals an odd integer.

b. Suppose evaluation of \( b_n \) requires time \( t_n \). Study of the code shows that for some constants \( a, b, \) and \( c \)

\[
t_0=t_1 =a, \\
2 \leq n \Rightarrow t_n = 2 \sum_{i=0}^{n-1} t_i + b_n + c.
\]

Eliminating the recursion one finds (and checks by induction on \( n \))

\[
2 \leq n \Rightarrow t_n=(4a+5b/2+c)3^{n-2}b/2,
\]

so that with \( A=(4a+5b/2+c)/9 \) and \( B=b/2 \) we have

\[
2 \leq n \Rightarrow t_n=A\cdot3^n-B.
\]

No other solution was received.