Computing Science 204 (B1)
Division on a Computer
Friday, July 21

Example. The following algorithm provides an approximate solution of the equation $bx = a$, where $a, b > 0$, using only the operations of addition, subtraction, and multiplication.

Since a binary computer can multiply and divide by a power of 2, we will assume that $\frac{1}{2} \leq b < 1$.

If we rewrite the equation as $x = (1-b)x + a$, and define $F(x) = (1-b)x + a$, then we see that $bx = a$ if and only if $F(x) = x$.

Now define a sequence $\{x_n\}_{n \geq 0}$ recursively as follows: $x_0 = F(0)$ and $x_{n+1} = F(x_n)$ for $n \geq 0$.

(a) Show that $\lim\limits_{n\to\infty} x_n = \dfrac{a}{b}$.

(b) Let $T(n)$ be the number of operations performed in the calcuation of x_n, find and solve a recurrence relation for $T(n)$.

proof. (a) We can compute the first few terms of the sequence $\{x_n\}_{n \geq 0}$ as follows:

$$
\begin{aligned}
x_0 &= F(0) = (1-b)0 + a = a \\
x_1 &= F(x_0) = (1-b)x_0 + a = (1-b)a + a \\
x_2 &= F(x_1) = (1-b)x_1 + a = (1-b)^2 a + (1-b)a + a \\
x_3 &= F(x_2) = (1-b)x_2 + a = (1-b)^3 a + (1-b)^2 a + (1-b)a + a \\
&\vdots \\
x_n &= F(x_{n-1}) = (1-b)x_{n-1} + a = (1-b)^n a + (1-b)^{n-1} a + \cdots + (1-b)^2 a + (1-b)a + a
\end{aligned}
$$

and an easy induction argument shows that

$$x_n = a \sum_{k=0}^{n} (1-b)^k$$

for all $n \geq 0$. Now, $\frac{1}{2} \leq b < 1$, so that $0 < 1 - b \leq \frac{1}{2}$, and therefore

$$\lim_{n\to\infty} x_n = a \lim_{n\to\infty} \sum_{k=0}^{n} (1-b)^k = a \lim_{n\to\infty} \frac{1-(1-b)^{n+1}}{1-(1-b)} = \frac{a}{b},$$

since $\lim\limits_{n\to\infty} (1-b)^{n+1} = 0$.

(b) We have $x_{n+1} = F(x_n) = (1-b)x_n + a$ for all $n \geq 0$.

If $T(n)$ is the number of operations performed in the calculation of x_n, then $T(n+1) = T(n) + 3$ for all $n \geq 0$, therefore,

$$T(n) = 3 + T(n-1) = 3 + 3 + T(n-2) = \quad \cdots \quad = \underbrace{3 + 3 + \cdots + 3}_{k \text{ terms}} + T(n-k),$$

so that when $k = n$, the initial condition $T(0) = 0$, gives $T(n) = 3n$ for all $n \geq 0$.