Two Exercises for Discussion

Shiyuan Deng

Department of Computer Science and Engineering
Chinese University of Hong Kong

Exercise 1 (Problem 1 of Regular Exercises List 1)

Let x be a real value. Define $\lfloor x \rfloor$ to be the largest integer that does not exceed x. For example, $\lfloor 2.5 \rfloor = 2$, whereas $\lfloor 3 \rfloor = 3$. Suppose that you are given an integer $n \geq 2$ in (a register of) the CPU. Write an algorithm to compute the value of $\lfloor \log_2 n \rfloor$ in no more than $100 \log_2 n$ time.

Exercise 2

You are given a positive integer n (that is stored in a register of the CPU). Design an algorithm to output the binary representation of n in no more than $100\lceil \log_2(n+1) \rceil$ time. For example, the binary representations of 7 and 8 are 111 and 1000, respectively.

Solution to Exercise 2

Let $b_i b_{i-1} \ldots b_0$ be the binary representation of n.

Observation 1: The integer division $\lfloor n/2 \rfloor$ gives $b_i b_{i-1} \ldots b_2 b_1$. Thus, the last bit b_0 can be calculated as $b_0 = n - \lfloor n/2 \rfloor \cdot 2$.

Observation 2: We can obtain b_1 by repeating the above on $b_i b_{i-1} \ldots b_2 b_1$.

Next, we analyze the running time. The binary form of n has $\lceil \log_2(n+1) \rceil$ digits. Therefore, we need to repeat for $\lceil \log_2(n+1) \rceil$ times. We leave it to you to implement each repeat using no more than 100 atomic operations (this is trivial). Therefore, the total cost is at most $100 \lceil \log_2(n+1) \rceil$.