HA2: Recurrence Relations

Daniel Detore
CS 385

October 2, 2024

1 p. 67, #4

1.a

$Mystery(n)$ computes the sum of the squares of all integers from 1 to n inclusive.

1.b

Its basic operation is addition.

1.c

Addition is executed n times for each call of $Mystery(n)$.

1.d

Its efficiency class is $\Theta(n)$.

1.e

Instead of the current strategy, calculating $Mystery(n) = \sum_{i=1}^{n} i^2$ with $\Theta(n)$, we can use my function $\text{𓅞}(n) = \frac{n(n+1)(2n+1)}{6}$ (only one calculation no matter the value of n) to enter efficiency class $\Theta(1)$. Let's prove this inductively. First, we can establish $\text{𓅞}(1) = Mystery(1)$ as $n = 1$ is the least value in the domain of $Mystery(n)$:

$$\begin{aligned} Mystery(1) &= \text{𓅞}(1) \\ \sum_{i=1}^{1} i^2 &= \frac{1(1+1)(2(1)+1)}{6} \\ 1^2 &= \frac{6}{6} \\ 1 &= 1 \end{aligned}$$

Then we can prove that $\text{𓅞}(n) = \text{𓅞}(n-1) + n^2$, which means that this formula is doing what it's supposed to do, i.e. summing the squares of the first n positive integers:

$$\begin{aligned}
\text{𓅞}(n) &= \text{𓅞}(n-1) + n^2 \\
&= \frac{(n-1)(n-1+1)(2(n-1)+1)}{6} + \frac{6n^2}{6} \\
&= \frac{(n-1)n(2n-1)+6n^2}{6} \\
&= \frac{2n^3+3n^2+n}{6} \\
&= \frac{n(n+1)(2n+1)}{6}
\end{aligned}$$

$\square$

2 p. 76, #1

2.a

Given $x(n) = x(n-1) + 5$ for $n > 1$, $x(1) = 0$:

1. Replace n by $n-1$: $x(n-1) = x(n-2) + 5$
   $\implies x(n) = (x(n-2)+5) + 5 = x(n-2) + 10$.
2. Replace n by $n-2$: $x(n-2) = x(n-3) + 5$
   $\implies x(n) = (x(n-3)+5) + 10 = x(n-3) + 15$.
3. We can generalize to $x(n) = x(n-i) + 5i$.
4. Initial condition $x(1) = 1 \implies n - i = 1 \implies i = n-1$.
5. Replace i: $x(n) = x(1) + 5(n-1) = 0 + 5(n-1)$.

Solution: $x(n) = 5(n-1)$.

2.b

Given $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 4$:

1. $x(n-1) = 3x(n-2)$
   $\implies x(n) = 3(3x(n-2)) = 3 * 3 * x(n-2)$.
2. Replace n by $n-2$: $3x(n-2) = 3x(n-3)$
   $\implies x(n) = 3(3(3x(n-3))) = 3 * 3 * 3 * x(x-3)$.
3. We can generalize to $x(n) = 3^i * x(n-i)$.
4. Initial condition $x(1) = 4 \implies n - i = 1 \implies i = n-1$.
5. Replace i: $x(n) = 3^{n-1} * x(n-(n-1)) = 3^{n-1} * x(1)$.

Solution: $x(n) = 3^{n-1} * 4$.

2.c

Given $x(n) = x(n-1) + n$ for $n > 0$, $x(0) = 0$:

1. $x(n-1) = x(n-2) + n - 1$
   $\implies x(n) = (x(n-2) + n - 1) + n = x(n-2) + 2n - 1$.
2. $x(n-2) = x(n-3) + n - 2$
   $\implies x(n) = (x(n-3) + n - 2) + 2n - 1 = x(n-3) + 3n - 3$.
3. We can generalize to $x(n) = x(n-i) + (i+1)n - i + 1$.
4. Initial condition $x(0) = 0 \implies n - i = 0 \implies i = n$.
5. Replace i: $x(n) = x(n-n) + (n+1)n - (n+1) = x(0) + n^2 + n - n + 1$.

Solution: $x(n) = n^2 + 1$.

2.d

Given $x(n) = x(n/2) + n$ for $n > 1$, $x(1) = 1$:

1. Replace n by 2^k: $x(2^k) = x(2^k/2) + 2^k = x(2^{k-1}) + 2^k$.
2. Replace 2^k by 2^{k-1}: $x(2^{k-1}) = x(2^{k-1}/2) + 2^{k-1} = x(2^{k-2}) + 2^{k-1}$
   $\implies x(2^k) = x(2^{k-2}) + 2^{k-1} + 2^k$.
3. We can generalize to $x(2^k) = 2^0 + 2^1 + 2^2 + \cdots + 2^k$. This summation is geometric which means $x(2^k) = \frac{2^{k+1}-1}{2-1} = 2^{k+1} - 1$.
4. Initial condition $x(1) = 1 \implies 2^{k-i} = n \implies k = \lg n$.
5. Replace k: $x(n) = 2^{\lg(n)+1} - 1 = 2n - 1$.

Solution: $x(n) = 2n - 1$.

2.e

Given $x(n) = x(n/3) + 1$ for $n > 1$, $x(1) = 1$:

1. Replace n by 3^k: $x(3^k) = x(3^k/3) + 1 = x(3^{k-1}) + 1$
2. Replace n by 3^{k-1}: $x(3^{k-1}) = x(3^{k-1}/3) + 1 = x(3^{x-2}) + 1$
   $\implies x(3^k) = (x(3^{x-2}) + 1) + 1$.
3. We can generalize to $x(3^k) = k + 1$.
4. Inital condition $x(1) = 1 \implies n = 3^k \implies k = \log_3(n)$
5. Replace k: $x(n) = \log_3(n) + 1$

3 p. 76-77, #3

3.a

Given $S(n) = S(n-1) + n^3$, $S(1) = 1$:

1. Replace n by $n-1$: $S(n-1) = S(n-2) + (n-1)^3$
   $\implies S(n) = S(n-2) + (n-1)^3 + n^3$.

2. Replace n by $n-2$: $S(n-2) = S(n-3) + (n-2)^3$
   $\implies S(n) = S(n-3) + (n-2)^3 + (n-1)^3 + n^3$.

3. We can generalize to
   $S(n) = S(n-i) + (n-i)^3 + (n-i+1)^3 + (n-i+2)^3 + \cdots + n^3 = S(n-i) + \sum_{k=0}^{i-1}(n-k)^3$.

4. Initial condition $x(1) = 1 \implies n - i = 1 \implies i = n - 1$.

5. Replace i: $S(n) = S(1) + \sum_{k=0}^{n-2}(n-k)^3 = 1 + \sum_{k=0}^{n-2}(n-k)^3$.

Solution: $S(n) = 1 + \sum_{k=0}^{n-2}(n-k)^3$. This means that the algorithm's basic operation is executed $n-1$ times.

3.b

This iterative (nonrecursive) algorithm and the recursive algorithm are the same in terms of running time as they both run at $\Theta(n)$.