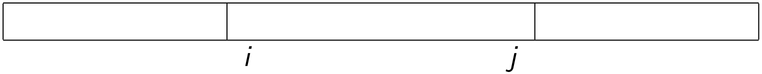
From *CLRS*, pg 39:

> *Describe a $\Theta(n \lg n)$-time algorithm that, given a set S of n integers and another integer x determines whether or not there exist two elements in S whose sum is exactly x.*



Invariant (Loop of `findPairSum`)

After $k \in \mathbb{W}$ iterations,

(a) $\forall\ a \in [0, i), s[a] + s[j] < x$

(b) $\forall\ b \in (j, n), s[i] + s[b] > x$

(c) $j - i = n - k - 1$

Correctness Claim (`findPairSum`)

The method $findPairSum$ returns two values in the given sequence that sum to x, if any exist.

Proof. *By induction on k, the number of iterations.*

Initialization. *Suppose $k = 0$ (before the loop starts). $i = 0$ and $j = n - 1$. The two ranges $[0, i)$ and (j, n) are empty, and so clauses (a) and (b) are vacuously true. Moreover, $j - i = n - 1 - 0 = n - 0 - 1 = n - k - 1$.*

Maintenance. *Suppose the invariant is true after k iterations, for some $k \geq 0$. Suppose a $k+1$st iteration occurs. By the guard (which must have been true), either $S[i] + S[j] < x$ or $S[i] + S[j] > x$.*

Suppose $S[i] + S[j] < x$. By the inductive hypothesis, for all $a \in [0, i)$, $S[a] + S[j] < x$. Hence for all $a \in [0, i+1)$, $S[a] + S[j] < x$. The invariant is maintained after i is incremented.

The situation is similar if $S[i] + S[j] > x$.

Additionally, either i is incremented or j is decremented. In either case $j_{new} - i_{new} = (j_{old} - i_{old}) - 1 = n - k - 1 - 1 = n - (k+1) - 1$.

Hence the invariant holds after $k+1$ iterations.

Termination. *After n iterations, $j - i = -1$ so $i > j$. Hence the loop will terminate after at most n iterations.*

After the loop terminates, either $i > j$ or $S[i] + S[j] = x$.

Suppose $i > j$. Then, by the loop invariant, no elements exist that sum to x, and the algorithm correctly returns `None`.

On the other hand, suppose $S[i] + S[j] = x$. Then the algorithm correctly returns $S[i]$ and $S[j]$. $\square$

2-3. Horner's rule for evaluating a polynomial:

$$\begin{aligned} P(X) &= \sum_{k=0}^{n} a_k x^k \\ &= a_0 + x(a_1 + x(a_2 + \cdots x(a_{n-1} + x a_n) \cdots)) \end{aligned}$$

a. $\Theta(n)$.
b. What's the naïve way? How naïve?

$y = 0$
$z = 1$
$i = 0$
while $i \leq n$
    $y = y + a_i \cdot z$
    $z = z \cdot x$
    $i = i + 1$

z keeps a running power of x. If we were computing x^n froms scratch, this would make each exponentiation $\Theta(n)$, so we would have $\Theta(n^2)$ total. (However, the book does say on pg 24 that we can assume exponentiation with small integer exponents are constant time.)
Both the given and my way are $\Theta(n)$. The difference is in the constant: 3 ops and 2 assignments vs 4 ops and 3 assignments.

For next time

Read Chapter 3, focusing on difference among formal definitions in Section 3.1

Do Problem 2-3.c. Prove the given invariant.
Do Exercises 3.1-(4 & 5)