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 $\mathtt{findPairSum}$ returns two values in the given sequence that sum to x, if any exist.

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

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, $j - i = n - 1 - 0 = n - 0 - 1 = n - k - 1$.*

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

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.

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

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 $S[j]$. $\square$

Proof of Horner's rule loop invariant:

Init. *After 0 iterations,* $y = 0$, $i = n$ *by assignment. So*

$$\sum_{k=0}^{n-(i+1)} a_{k+i+1} = \sum_{k=0}^{-1} a_{k+i+1}x^k = 0 = y$$

Maint. *Now, suppose this holds true after* N *iterations, that is*

$$y_{old} = \sum_{k=0}^{n-(i_{old}+1)} a_{k+i_{old}+1}x^k$$

where y_{old} *and* i_{old} *are* y *and* i *after* N *iterations. Likewise, let* y_{new} *and* i_{new} *be the values after* $N + 1$ *iterations.*

By assignment $i_{new} = i_{old} - 1$. *Then*

$$
\begin{aligned}
y_{new} &= a_{i_{old}} + x \cdot y_{old} && \textit{by assignment} \\
&= a_{i_{old}} + x \cdot \sum_{k=0}^{n-(i_{old}+1)} a_{k+i_{old}+1} x^k \\
&= a_{inew-1} + x \cdot \sum_{k=0}^{n-(inew+2)} a_{k+inew} x^k && \textit{by substitution} \\
&= a_{inew-1} + \sum_{k=0}^{n-(inew+2)} a_{k+inew} x^{k+1} && \textit{by distribution} \\
&= a_{inew-1} + \sum_{k=1}^{n-(inew+1)} a_{k+inew+1} x^k && \textit{by change of variables} \\
&= a_{0+inew-1} x^0 + \sum_{k=1}^{n-(inew+1)} a_{k+inew+1} x^k \\
&= \sum_{k=0}^{n-(inew+1)} a_{k+inew+1} x^k && \square
\end{aligned}
$$