Prolegomena unit outline:

- Algorithms and correctness (Wednesday and **today**)
- Algorithms and efficiency (all next week)
- Abstract data types (Mon, Jan 27)
- Data Structures (Jan 29 and 31)

Today:

- The "Binary search" problem
- Class invariants (`LinkedList`)

What good are invariants?

- They are a tool for reasoning about the state and progress of an algorithmic process
- They are a way to explain the meaning of a variable and capture how the variables relate to each other.
- They help with testing and debugging.
- They are a means for proving that an algorithm is correct.

Given a list `sequence` and a total order, determine whether `sequence` is sorted by the given total order.

Given a list `sequence` sorted by a given total order `TO` and given an `item`, return

$$\begin{cases} -1 & \text{if } \forall\ i \in [0, n), \texttt{sequence}[i] \neq \texttt{item} \\ k & \text{otherwise, where } \texttt{sequence}[k] = \texttt{item} \end{cases}$$

Given a list `sequence` sorted by a given total order `TO` and given an `item`, return

$$\begin{cases} -1 & \text{if } \forall\, i \in [0, n), \texttt{sequence}[i] \neq \texttt{item} \\ k & \text{otherwise, where } \texttt{sequence}[k] = \texttt{item} \end{cases}$$

Invariant (Loop of `binary_search`.)

(a) *If* $\exists\, j \in [0, n)$ *such that* $\texttt{item} = \texttt{sequence}[j]$, *then* $\exists\, j \in [\texttt{low}, \texttt{high})$ *such that* $\texttt{item} = \texttt{sequence}[j]$.

(b) *After* i *iterations*, $\texttt{high} - \texttt{low} \leq \frac{n}{2^i}$.

(a) If $\exists\ j \in [0, n)$ such that $\texttt{item} = \texttt{sequence}[j]$, then $\exists\ j \in [\texttt{low}, \texttt{high})$ such that $\texttt{item} = \texttt{sequence}[j]$.

(b) After i iterations, $\texttt{high} - \texttt{low} \le \frac{n}{2^i}$.

Initialization.

(a) Initially $\texttt{low} = 0$ and $\texttt{high} = n$, so the hypothesis and conclusion are identical.

(b) No iterations yet, so

$$\texttt{high} - \texttt{low} = n - 0 = n = \frac{n}{1} = \frac{n}{2^0}$$

(a) If $\exists\ j \in [0, n)$ such that $\texttt{item} = \texttt{sequence}[j]$,
then $\exists\ j \in [\texttt{low}, \texttt{high})$ such that $\texttt{item} = \texttt{sequence}[j]$.

(b) After i iterations, $\texttt{high} - \texttt{low} \leq \frac{n}{2^i}$.

Maintenance. Distinguish $\texttt{low}_{\text{pre}}$ and $\texttt{low}_{\text{post}}$, $\texttt{high}_{\text{pre}}$ and $\texttt{high}_{\text{post}}$. Let i be the number of iterations completed. We're given that if $\exists\ j \in [0, n)$ such that $\texttt{item} = \texttt{sequence}[j]$, then $\exists\ j \in [\texttt{low}_{\text{pre}}, \texttt{high}_{\text{pre}})$ such that $\texttt{item} = \texttt{sequence}[j]$; also that $\texttt{high}_{\text{pre}} - \texttt{low}_{\text{pre}} \leq \frac{n}{2^{i-1}}$ (this is our *inductive hypothesis*). The guard also assures us that $\texttt{high}_{\text{pre}} - \texttt{low}_{\text{pre}} > 1$.
We have three possibilities, corresponding to the if-elif-else:

(a) If $\exists\ j \in [0, n)$ such that $\texttt{item} = \texttt{sequence}[j]$,
then $\exists\ j \in [\texttt{low}, \texttt{high})$ such that $\texttt{item} = \texttt{sequence}[j]$.

(b) After i iterations, $\texttt{high} - \texttt{low} \leq \frac{n}{2^i}$.

Case 1: Suppose $\texttt{item} < \texttt{sequence}[\texttt{mid}]$.

(a) Since sequence is sorted, $\forall\ j \in [\texttt{mid}, \texttt{high}_{\text{pre}})$, $\texttt{item} < \texttt{sequence}[j]$. Thus if $\exists\ j \in [\texttt{low}_{\text{pre}}, \texttt{high}_{\text{pre}})$, then $\exists\ j \in [\texttt{low}_{\text{pre}}, \texttt{mid})$, that is (with the update to high but not to low), $\exists\ j \in [\texttt{low}_{\text{post}}, \texttt{high}_{\text{post}})$
Now, by transitivity of the conditional, if $\exists\ j \in [0, n)$ such that $\texttt{item} = \texttt{sequence}[j]$, then $\exists\ j \in [\texttt{low}_{\text{post}}, \texttt{high}_{\text{post}})$ such that $\texttt{item} = \texttt{sequence}[j]$.

(b) If the length of the range is odd, then the sub-ranges above and below mid are of equal size, each half of the range length minus one. If the range length is even, then the lower subrange is half that size and the upper subrange is one less than half. Either way we throw away at least half and keep no more than half. So,

$$\texttt{high}_{\text{post}} - \texttt{low}_{\text{post}} \leq \frac{1}{2} \cdot (\texttt{high}_{\text{pre}} - \texttt{low}_{\text{pre}}) \leq \frac{1}{2} \cdot \frac{n}{2^{i-1}} \leq \frac{n}{2^i}$$

(a) If $\exists\, j \in [0, n)$ such that $\texttt{item} = \texttt{sequence}[j]$,
then $\exists\, j \in [\texttt{low}, \texttt{high})$ such that $\texttt{item} = \texttt{sequence}[j]$.

(b) After i iterations, $\texttt{high} - \texttt{low} \leq \frac{n}{2^i}$.

Case 2: Suppose $\texttt{item} = \texttt{sequence}[\texttt{mid}]$.

(a) Immediately we have $\exists\, j \in [\texttt{mid}, \texttt{mid}+1)$, and, with the update to high and low, that means $\exists\, j \in [\texttt{low}_{\text{post}}, \texttt{high}_{\text{post}})$. Moreover, the conditional is $T \to T \equiv T$.

(b) Note $\texttt{high}_{\text{post}} - \texttt{low}_{\text{post}} = 1$. Earlier we said $1 < \texttt{high}_{\text{pre}} - \texttt{low}_{\text{pre}} \leq \frac{n}{2^{i-1}}$.
Since $\texttt{high}_{\text{pre}} - \texttt{low}_{\text{pre}}$ must be a whole number, $2 \leq \frac{n}{2^{i-1}}$, and so $1 \leq \frac{n}{2^i}$.
Finally $\texttt{high}_{\text{post}} - \texttt{low}_{\text{post}} \leq \frac{n}{2^i}$.

Case 3: Suppose $\texttt{item} > \texttt{sequence}[\texttt{mid}]$. This is similar to Case 1. □

Correctness claim (`binary_search`.)

After at most $\lg n$ iterations, `binary_search` returns as specified.

Proof. Suppose $i \geq \lg n$. Then $2^i \geq n$ and $\frac{n}{2^i} \leq 1$. Hence $\texttt{high} - \texttt{low} \leq 1$ and the guard fails.

Invariant 2.a still means that if the item is anywhere, it's in the range. The guard implies that on loop exit the range has size 0 or 1.

Suppose the range has size 0. Then the item isn't in the range (since nothing is), and thus it isn't anywhere. Since $\texttt{high} = \texttt{low}$, the first part of the conditional fails and and -1 is returned, as specified.

On the other hand, suppose the range has size 1. We still don't know if the item is in the range, but we have only one location to check. If it's in `sequence[low]`, then we return `low`, which meets the specification. Otherwise the second part of the condition fails and -1 is returned, as specified. □

Invariant (Loop of `binary_search`.)

(a) *If* $\exists\, j \in [0, n)$ *such that* $\texttt{item} = \texttt{sequence}[j]$, *then* $\exists\, j \in [\texttt{low}, \texttt{high})$ *such that* $\texttt{item} = \texttt{sequence}[j]$.

(b) *After* i *iterations*, $\texttt{high} - \texttt{low} \leq \frac{n}{2^i}$.

Invariant (Preconditions of `binary_search_recursive`)

(a) *If* $\exists\, j \in [0, n)$ *such that* $\texttt{item} = \texttt{sequence}[j]$, *then* $\exists\, j \in [\texttt{low}, \texttt{high})$ *such that* $\texttt{item} = \texttt{sequence}[j]$.

(b) $\texttt{low} \leq \texttt{high}$

Invariant (Class `LinkedList`)

(a) $\texttt{head} = \texttt{null}$ *iff* $\texttt{tail} = \texttt{null}$ *iff* $\texttt{size} = 0$.

(b) *If* $\texttt{tail} \neq \texttt{null}$ *then* $\texttt{tail.next} = \texttt{null}$.

(c) *If* $\texttt{head} \neq \texttt{null}$ *then* $\mathtt{tail}$ *is reached by following* $\texttt{size} - 1$ $\mathtt{next}$ *links from* $\mathtt{head}$.

Coming up:

Finish the **pretest project**, *due Tues, Jan 21*

Due **Thursday Jan 21** *(end of day):*
Read Section 1.2 (long section—spread it out)
Do Exercise 1.(6)—submit through Canvas
Take quiz (algorithms and correctness)

Due **Friday, Jan 24** *(end of day)*
Read Sections 1.(3 & 4) (also long—spread it out)
Do exercises 1.(17 & 18)
Take quiz