

## Prolegomena unit outline:

- ▶ Algorithms and correctness (Wednesday and **today**)
- ▶ Algorithms and efficiency (all next week)
- ▶ Abstract data types (Mon, Jan 22)
- ▶ Data Structures (Jan 24 and 26)

## Today:

- ▶ Finish “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` sorted by a given total order `TO` and given an `item`, return

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

Invariant (Loop of `binary_search`.)

- (a) *If  $\exists j \in [0, n)$  such that  $\text{item} = \text{sequence}[j]$ , then  $\exists j \in [\text{low}, \text{high})$  such that  $\text{item} = \text{sequence}[j]$ .*
- (b) *After  $i$  iterations,  $\text{high} - \text{low} \leq \frac{n}{2^i}$ .*

- (a) If  $\exists j \in [0, n)$  such that  $\text{item} = \text{sequence}[j]$ ,  
then  $\exists j \in [\text{low}, \text{high})$  such that  $\text{item} = \text{sequence}[j]$ .
- (b) After  $i$  iterations,  $\text{high} - \text{low} \leq \frac{n}{2^i}$ .

### Initialization.

- (a) Initially  $\text{low} = 0$  and  $\text{high} = n$ , so the hypothesis and conclusion are identical.
- (b) No iterations yet, so

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

- (a) If  $\exists j \in [0, n)$  such that  $\text{item} = \text{sequence}[j]$ ,  
then  $\exists j \in [\text{low}, \text{high})$  such that  $\text{item} = \text{sequence}[j]$ .
- (b) After  $i$  iterations,  $\text{high} - \text{low} \leq \frac{n}{2^i}$ .

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

We have three possibilities, corresponding to the if-elif-else:

- (a) If  $\exists j \in [0, n)$  such that  $\text{item} = \text{sequence}[j]$ ,  
then  $\exists j \in [\text{low}, \text{high})$  such that  $\text{item} = \text{sequence}[j]$ .
- (b) After  $i$  iterations,  $\text{high} - \text{low} \leq \frac{n}{2^i}$ .

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

- (a) Since  $\text{sequence}$  is sorted,  $\forall j \in [\text{mid}, \text{high}_{\text{pre}})$ ,  $\text{item} < \text{sequence}[j]$ . Thus if  $\exists j \in [\text{low}_{\text{pre}}, \text{high}_{\text{pre}})$ , then  $\exists j \in [\text{low}_{\text{pre}}, \text{mid})$ , that is (with the update to  $\text{high}$  but not to  $\text{low}$ ),  $\exists j \in [\text{low}_{\text{post}}, \text{high}_{\text{post}})$   
Now, by transitivity of the conditional, if  $\exists j \in [0, n)$  such that  $\text{item} = \text{sequence}[j]$ , then  $\exists j \in [\text{low}_{\text{post}}, \text{high}_{\text{post}})$  such that  $\text{item} = \text{sequence}[j]$ .
- (b) If the length of the range is odd, then the sub-ranges above and below  $\text{mid}$  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,

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

- (a) If  $\exists j \in [0, n)$  such that  $\text{item} = \text{sequence}[j]$ ,  
then  $\exists j \in [\text{low}, \text{high})$  such that  $\text{item} = \text{sequence}[j]$ .
- (b) After  $i$  iterations,  $\text{high} - \text{low} \leq \frac{n}{2^i}$ .

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

- (a) Immediately we have  $\exists j \in [\text{mid}, \text{mid} + 1)$ , and, with the update to  $\text{high}$  and  $\text{low}$ , that means  $\exists j \in [\text{low}_{\text{post}}, \text{high}_{\text{post}})$ . Moreover, the conditional is  $T \rightarrow T \equiv T$ .
- (b) Note  $\text{high}_{\text{post}} - \text{low}_{\text{post}} = 1$ . Earlier we said  $1 < \text{high}_{\text{pre}} - \text{low}_{\text{pre}} \leq \frac{n}{2^{i-1}}$ . Since  $\text{high}_{\text{pre}} - \text{low}_{\text{pre}}$  must be a whole number,  $2 \leq \frac{n}{2^{i-1}}$ , and so  $1 \leq \frac{n}{2^i}$ . Finally  $\text{high}_{\text{post}} - \text{low}_{\text{post}} \leq \frac{n}{2^i}$ .

**Case 3:** Suppose  $\text{item} > \text{sequence}[\text{mid}]$ . 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 `high` - `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 `high` = `low`, the first part of the conditional fails 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.  $\square$



## Invariant (Loop of `binary_search`.)

- (a) *If  $\exists j \in [0, n)$  such that `item = sequence[j]`, then  $\exists j \in [\text{low}, \text{high})$  such that `item = sequence[j]`.*
- (b) *After  $i$  iterations,  $\text{high} - \text{low} \leq \frac{n}{2^i}$ .*

## Invariant (Preconditions of `binary_search_recursive`)

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

## Invariant (Class LinkedList)

- (a)  $\text{head} = \text{null}$  iff  $\text{tail} = \text{null}$  iff  $\text{size} = 0$ .
- (b) If  $\text{tail} \neq \text{null}$  then  $\text{tail.next} = \text{null}$ .
- (c) If  $\text{head} \neq \text{null}$  then *tail* is reached by following  $\text{size} - 1$  next links from *head*.

## Coming up:

*Due **Tuesday Jan 16** (class time)*

*Read Section 1.2 (long section—spread it out)*

*Do Exercises 1.6*

*Take quiz*