CSE 331
Software Design & Implementation

Winter 2026

Section 8 – Arrays

Administrivia

- HW7 released tonight - **Due @ 11:59pm next Wed**

Array Notation: Indexing

- Arrays are treated like Lists *mathematically*
  - They are an alternative way to represent Lists in *code*

- To get the jth element, use at(L, j), abbreviated "L[j]"

$$\begin{aligned} at &: (List, \mathbb{N}) \rightarrow \mathbb{Z} \\ at(nil\ ,\ n) &:= \text{undefined} \\ at(x :: L\ ,\ 0) &:= x \\ at(x :: L\ ,\ n+1) &:= at(L, n) \end{aligned}$$

  - Must ensure that 0 <= j < length of array

More Array Notation

- **Prefix:** All elements from the start to j+1 (exclusive):
  - A[..j] = prefix(A, j+1)
    - **Add note about this not matching Python**

- **Suffix:** All elements starting at j (inclusive) to the end:
  - A[j..] = suffix(A, j)

- Thus, A[..j-1] ++ A[j..] = A
  - 331 notation is inclusive of both indices

Other useful facts:

- A[len(A)..] = nil
- A[..-1] = nil
- A[..j-1] ⧺ A[j] = A[..j]
- A[..len(A)-1] = A

‘For Any’ Facts

- Necessary facts about arbitrary parts of an array
- Ex: To show an array is sorted in asc formally:
  - A[j] < A[j+1] *for any* 0 ≤ j < len(A) – 1

```
// @requires A[i] < A[i+1] for any 0 <= i < len(A)-1
// @returns false if A[i] != y for any 0 <= i < len(A)
//          true  otherwise
public boolean bsearch(int[] A, int y) { … }
```

Array Mutation

- Array mutation can change “for any” facts!
- Ex:

{{ A[j] < A[j+1] for any $0 \le j \le 9$ }}

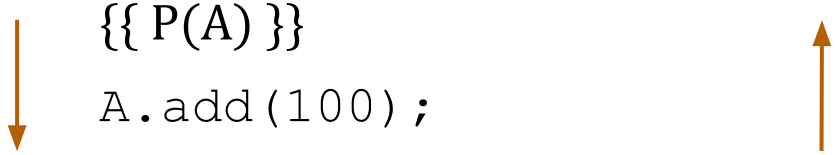
```
A[0] = 100;
```

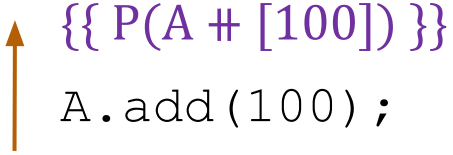
{{ (A[j] < A[j+1] for any $\mathbf{1} \le j \le 9$) and A[0] = 100 }}

- Old facts about A[0] could be invalidated!
  - Need to update the range of “for any” facts

Mutating Arrays (add/remove)

- Adding to the end of an array





- Removing from the end of an array