
CSE 331

Software Design & Implementation

Winter 2026

Section 5 – Mutable ADTs

Administrivia

- HW5 released tonight - **Due @ 11:59pm Wednesday**



Externally Documenting ADTs - Review

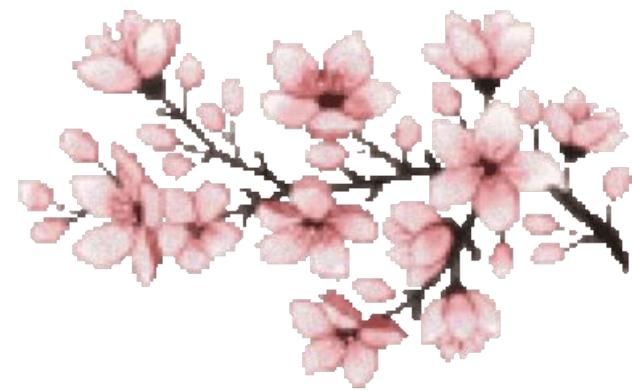
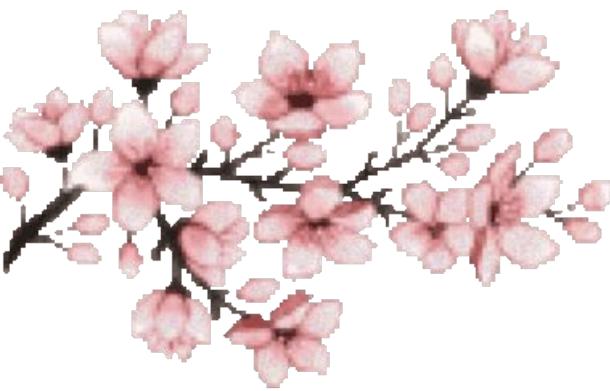
- JavaDoc comments in interface use “tags” to describe what ADT methods do in terms of the **abstract state**

```
/**  
 * High level description of what function does  
 * @param a What "a" represents + any conditions  
 * @requires Rules about multiple params and Abstract State (obj)  
 * @returns Detailed description of return value  
 * @throws Condition when errors will be thrown  
 */
```

- For mutable ADTs, will have 2 additional tags to describe “mutator” methods
 - * **@modifies** states what *could* be mutated by function (obj)
 - * **@effects** Detailed description of *guaranteed* changes

Specification Strength Mutation - Review

- Adding more to `@returns` **increases** specification strength
- Adding more to `@effects` **increases** specification strength
- Adding more to `@modifies` **decreases** specification strength
 - Note that this is not a promise or guarantee



Testing - Review

- **Statement Coverage**
 - Test every executable statement reachable by an allowed input
- **Branch Coverage**
 - For every conditional, test all branches for allowed inputs
- **Loop Coverage**
 - Every loop/recursive call must be tested on 0, 1, any 2+ iterations for allowed inputs
- **Exhaustive Testing**
 - Test all possible inputs for functions with ≤ 10 allowed inputs

Mutable ADT Testing

- **Mutable ADTs**

- Must also test that mutated values were updated correctly.
- Test state before and after each mutator method.

