CSE 250 Recitation

February 5 - 6: PA1, Asymptotic Analysis

PA1: Getting Started

- **PA1** revolves around linked lists and how to implement them
- All PAs this semester start by writing tests
- Why Test Driven Development?
  - Deepens your understanding of the problem
  - Enables you to test your code without submitting to Autolab
  - Writing code before thinking about the problem will lead to disaster

PA1: Getting Started

- Remember when writing tests, **understanding the expected behavior** of each method is more important than how to make your implementation
- Ask “**what situations could break my code**”
- Let's try to come up with some good linked lists for testing
  - **Side note:** how can we make these lists without relying on methods like insert

PA1: SortedList Exercise

Discussion: What are the features of `SortedList` in PA1 that make it different from a vanilla LinkedList?

PA1: SortedList Exercise

Discussion: What are the features of **`SortedList`** in PA1 that make it different from a vanilla LinkedList?

- Doubly Linked
- Sorted
- Duplicate values are stored in a single linked list node
  - Each node therefore stores the **value** AND the **count**
- Some methods take a node reference as a hint

PA1: SortedList Exercise

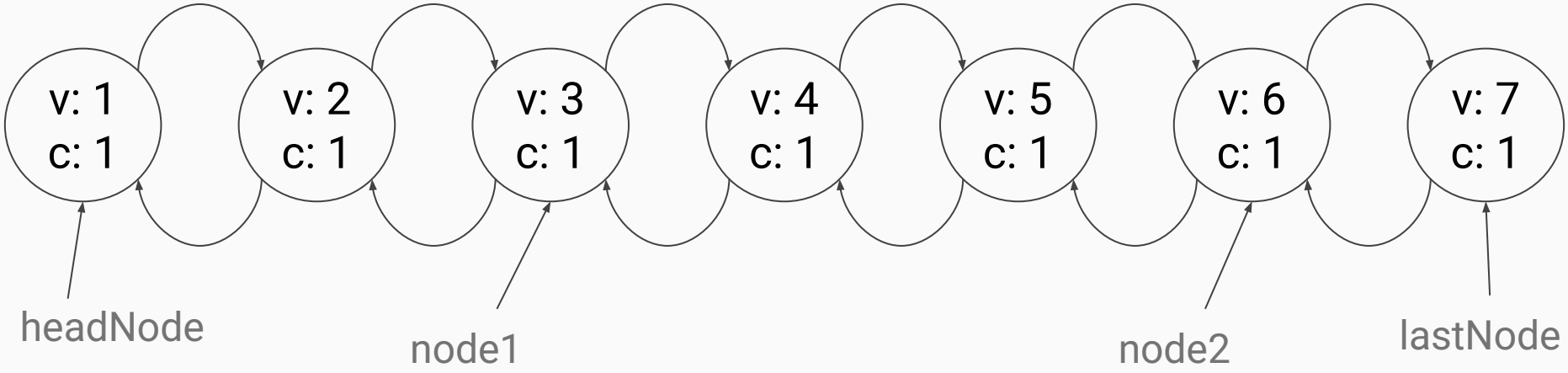
Exercise: Write a sequence of 7 values and draw the `SortedList` that would result from inserting those 7 values into an initially empty `SortedList`

Below your list write out the following questions (don't answer them)
(Pick X, Y, node1 and node2 that you think make them **difficult to answer**)

1. Is this a valid `SortedList` of length 7?
2. What value is returned by `get(X)`?
3. What node is returned by `findRefBefore(Y, node1)`?
4. What does this list look like after calling `remove(node2)`?

An Example (Not Necessarily a Good One)

Insertions: 1, 2, 3, 4, 5, 6, 7



1. Is this a valid list of length 7?
2. What value is returned by get(0)?
3. What node is returned by findRefBefore(4, node1)
4. What does the list look like after calling remove(node2)?

PA1: SortedList Exercise Part 2

Exercise: Trade papers with another student in the class

1. Answer the questions posed by the other student about their list
    a. If you believe their list is not a valid `SortedList` of length 7, explain why
2. Below their questions, state *at least* one scenario that their list/questions may not account for
    a. Example for the previous slide: It does not have any nodes with count > 1
3. Switch back and check the other students work

PA1: SortedList Exercise Wrap Up

Wrap Up: Think about this exercise when writing tests

1. Take the role of the other student finding holes in your tests to think about things you may not be checking for
2. To cover the holes you may need to make different lists, or may just need to ask different questions about your current lists
    a. The more situations you cover, the more confident you can be in an implementation that passes your tests
3. **DON'T STOP** writing tests after the testing phase…keep adding tests

Other Tips for Testing

- Try to test just one function at a time
  - What if we want to test get? How can we build a list without using insert?
- After writing some tests, re-read the handout and for each function be on the lookout for sentences that describe untested behavior
- **If AutoLab finds a bug in your implementation, don't think "I need to fix that bug", think "I need to figure out why MY tests didn't catch it".**

Asymptotic Analysis

Proving Bounds

Let $g(n) = 3n + n^2$. Prove that $g(n) \in O(n^2)$, $g(n) \in \Omega(n^2)$

First...what is the definition of big-O?

Proving Bounds

Let $g(n) = 3n + n^2$. Prove that $g(n) \in O(n^2)$, $g(n) \in \Omega(n^2)$

First...what is the definition of big-O?

$g(n) \leq c\, f(n)$, **for all** $n \geq n_0$ for some $c > 0$, and $n_0 \geq 0$

What is the definition of big-Ω?

Proving Bounds

Let $g(n) = 3n + n^2$. Prove that $g(n) \in O(n^2)$, $g(n) \in \Omega(n^2)$

First...what is the definition of big-O?

$g(n) \leq c\, f(n)$, **for all** $n \geq n_0$ for some $c > 0$, and $n_0 \geq 0$

What is the definition of big-Ω?

$g(n) \geq c\, f(n)$, **for all** $n \geq n_0$ for some $c > 0$, and $n_0 \geq 0$

Proving Bounds

Exercise: Prove the following

1. $g(n) \in O(n^2)$
2. $g(n) \in \Omega(n^2)$
3. $g(n) \in \Theta(n^2)$
4. $T(n) \in O(n^3)$
5. $T(n) \in \Omega(n^2)$

$$g(n) = 3n + n^2$$

$$T(n) = n^2 + 5n + 28 + \begin{cases} 3n^3 + 4n + 5 & \text{if true} \\ 4n + 5 & \text{otherwise} \end{cases}$$

Hint: For T, consider the fact that it is either:

$$3n^3 + n^2 + 9n + 33 \quad \text{OR} \quad n^2 + 9n + 33$$

More Examples

Prove the following:

$12 \log(10 \times 2^n) \in O(n)$

$n^2 + n \log(n) \in O(2^n)$

$n^2 + 15n^3 \in \Omega(n)$

$$\sum_{i=1}^{n} i \in \Omega(n^2)$$