

CSC207.01 2014S, Class 30: Linked Lists in Java

Overview

- Preliminaries.
 - Admin.
- Linked lists.
- Implementation details.
- Lab

Preliminaries

Admin

- No reading for tomorrow.
- No writeup for today.
- Exam 2 to be distributed on Wednesday (assuming I get it written).
- Mentor session tonight at 7:00 p.m.
- Extra credit:
 - Lunchtime meeting today alt careers in education JRC 225

Questions on yesterday's lab?

Linked lists

Once we've decided on an ADT, we think about the implementation

- "Data structure" \approx "implementation"
- Reminder: ADT
 - Philosophy: What should it do?
 - Also things like mutability
 - Use cases: What are ways in which we'll use it?
 - Methods: What methods does it provide?
- N important questions for data structures
 - What is our goal? To implement the given ADT (lists)
 - How do we structure the data / lay out memory?
 - Array-based lists: Big chunk of memory ($N \times \text{sizeof}(\text{reference})$)
 - The reference to item i is at $\text{base} + i * \text{sizeof}(\text{reference})$
 - Given that decision, how do we implement the methods?
 - next: Increment i , grab the value from the old position i

- add: Shove to the right, insert value
- remove: Shove to the left
- Given those implementations, what are the running times?
 - next: $O(1)$
 - add: $O(n)$
 - remove: $O(n)$
- Can we make the add/remove methods faster?
- Strategy: Linked lists
 - Structure the data
 - Node: Stores the value and a link to the next element
 - The enclosing list stores the front (or a dummy node before the front)
 - Implement the methods
 - Iterator keeps track of the PREVIOUS node
 - next: `it.current = it.current.next; return it.current.data;`
 - add: `it.current.next = new Node(value,it.current.next); advance iterator`
 - remove: cut out node
 - prev: start at the beginning and advance until you get the previous element
 - Analyze
 - next: $O(1)$
 - add: $O(1)$
 - remove: $O(1)$
 - prev: $O(n)$

Implementation details

Lab

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.