

More Linked Lists

CSSE 221

Fundamentals of Software Development Honors

Rose-Hulman Institute of Technology

Announcements

- LinkedListBasic test bug:
 - In test_reverseRecursive, change reverse to reverseRecursive!
- All C Projects due Friday 5:00 pm
- Final Exam Monday night, 6-10 pm, here and next door.

Feedback on feedback...

- Thanks much for your thoughtful feedback on the Java portion of the course!

Minoring in CSSE

- Enjoying this class?
- Is your primary interest outside of CSSE, but wondering how you can do more computing?
- You are already $2/7$ of the way there

Courses on the horizon

CS MINOR

CSSE230: Data Structures and Algorithm Analysis (W/S)
CSSE432: Networks (S)
CSSE241: Computing in a Global Soc. (F)
CSSE413: AI (F)
CSSE351: Graphics (F; with DE1)
CSSE481: Web-based info (w/ 230)
CSSE461: Computer Vision (S, w/ DE1)
CSSE463: Image Recognition (W, w/ DE1)
With MA275 (Discrete Math):
CSSE333: Databases (W, w/230)
CSSE304: Prog Lang Concepts (S, 230)
CSSE/MA479: Cryptography (S)
With ECE130:
CSSE232: Computer Architecture
CSSE332: Operating Systems

SE MINOR

CSSE230: Data Structures and Algorithm Analysis
CSSE 371 Software Requirements and Specification (F)
CSSE 372 Software Project Management (F)
CSSE 374 Software Architecture and Design (W)
CSSE 375 Software Construction and Evolution (S)
CSSE 376 Software Quality Assurance (S)
CSSE 377 Software Architecture and Design II (F)
With MA275:
CSSE 373 Formal Methods in Specification and Design

SVN in Future

- I will blow away the public repository and all your team repositories sometime before I teach 221 again, so as to start with a clean slate (in actuality, I plan to archive everything).
- However, I will leave your personal repository, so you can use for other classes.
- (You can also request by email to keep team repos.)
- Furthermore, you can ask a CS prof to make you an SVN repos for any other team at any time
 - I routinely make them for ME senior projects

Final Exam

- Same rules as previous exams:
- A closed-resource portion
 - Part may possibly be on Angel
- An open-resource version
 - Coding C in your IDE

Topics

Heavy emphasis:

1. Data structures
 1. Arrays, Lists, Stacks, Queues, Sets, and Maps
2. Sorting (3 methods) and searching (seq. and binary)
3. Big-oh efficiency
 1. loops
 2. data struct operations
 3. sorting and searching
 4. using sorted data
4. Recursion
5. C language
6. Linked Lists

Other things you should know. It's a comprehensive exam, so everything is fair game:

1. Unit testing
2. UML
3. Func. Objects/Comparators
4. Inheritance and polymorphism
5. Iterators
6. Threads and animation
7. GUIs
8. File I/O

Live Coding

- Will be all in C
 - Approximately 25-35% of the exam
 - Lots of linked lists
 - Some strings
 - Recursion (perhaps on a linked list or with strings?)
- Some code-writing on paper part.

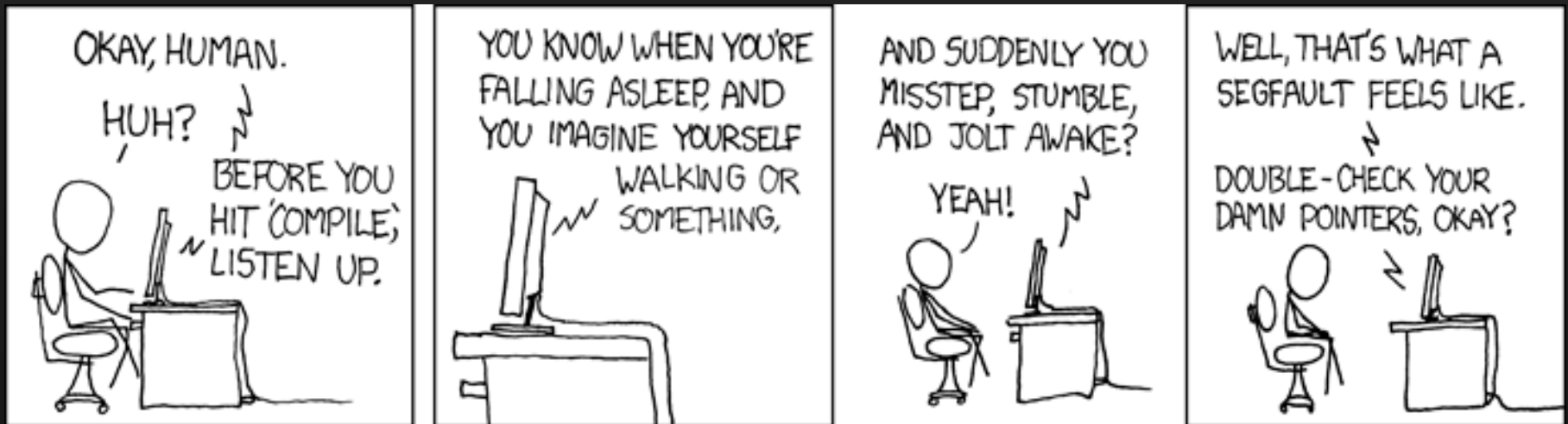
Questions?

- Come to class tomorrow with questions

Questions on Basic LinkedLists?

- Have you handled all 3 cases (empty list, 1-node list, and 2+ node list) in each function?
 - Use the test script to find out
- Have you freed every node you've deleted?
 - Do a quick visual check of all your delete functions.

<http://www.xkcd.com/371/>



Eliminating Special Cases

- Head and tail nodes:
 - Head: an extra node at the beginning of the linked list implementation that points to the node containing the first List item. The contents of the head node are **not** part of the List. This is stored in the list instead of “first”
 - Tail: an extra node at the back of the list, for symmetry in doubly-linked lists.
- Thus there are two nodes in the representation of the empty list, three nodes in the representation of a one-element list, etc.

An enhanced version

- Once you are done, you should check out the `LinkedListEnhanced` project. It includes the following enhancements:
 - Doubly-linked (helps `removeLast` and `removeAt`)
 - Dummy nodes (head and tail) to remove special cases.
 - Size field to make `getSize` $O(1)$.