CSC207.01 2014S, Class 33: An Introduction to Linear Structures

Overview

- Preliminaries.
  - Upcoming work.
  - Admin.
  - Questions on the exam.
- Today's concepts.
  - Linear structures.
  - Stacks.
  - An application of stacks: Matching parens.
- Lab.

Preliminaries

Upcoming work

- Today's writeup: Exercise 4.
- Reading for Tuesday: Queues.
- Exam 2 due Friday in class.

Admin

- I am returning the makeups for exam 1.
  - 4 of you increased your grade
  - 8 of you had no change. Many of your average functions still had bugs, many of your replacement functions were still O(n^2).
- Review session Tuesday night at 7 (we think)
- CS study break tonight (I think)
- Consider running for the SEPC
- Get ready to vote on T-Shirts (the Curried Lamb-Duh T-shirt is clearly the best)

Extra Credit

- Convocation, Wednesday, noon: Elizabeth Colbert (the Sixth Extinction?)
- CS extras Thursday: ???
- CS table Friday: TBD
- Dance Troop performances in Chicago this weekend.

○ Double extra credit, because it involves figuring out where

Questions on the Exam

Will you write some unit tests for the filtered list `remove` *method?*

I'll try.

Linear structures

New ADT: Linear Structure

- What is the big purpose/philosophy?
  - ○ A very simple mutable collection
- What methods should it have?
  - ○ Add something
  - ○ Get something
- Use cases: What problems would it be useful for?
  - ○ Sam's office
  - ○ To do list, job jar, ...

Interesting issue: We create different ADTs by specifying *policies* for the get method.

- Random: Give me something unpredictable.
- LIFO: Give me the most recently added thing still in the stucture
- FIFO: Give me the least recently added thing still in the structure
- Priority: Give me the highest priority (e.g., smallest, largest) thing

Four policies -> Four new ADTs.

Stacks

- What is the philosophy:
  - ○ A simple linear structure with a LIFO policy
- Methods
  - ○ put(T val)
  - ○ T get() - get/remove most recently added value that still remains
- Use cases:
  - ○ Matching. When you see an opening value, push on the stack. When you see a closing value, pop it off the stack.
  - ○ Function calls in almost any modern language.
  - ○ Model portions of a cafeteria

Lab