

CSC207.01 2014S, Class 46: Pause for Breath

Overview

- Preliminaries.
 - Admin.
 - Upcoming work.
 - Extra credit.
 - Questions.
- Project stuff.
 - Possible extensions to the project.
 - More notes on submitted work.
 - Planning presentations.
- Implementing heaps.
- Heapsort.
- Doing better?

Preliminaries

Admin

- Sugar!
- Mentor session tonight at 7 pm in 3821.
- It's not clear if I'll have time to review any of the other projects. We'll continue to talk about code and other issues.
- I need two graders for next semester who can do detailed comments. Yes, I will provide training.
- Please download the heap lab code (particularly if there's time at the end of class). <https://github.com/Grinnell-CSC207/heaps>

Upcoming Work

- Reading for Wednesday: No reading.
- Today's writeup: No writeup.
- Part 2 of the project is due Friday at 10:30 p.m. (No, I'm not updating the Web page.)

Extra Credit

- One pride week activity.
- Math extra Thursday: Sphere Packing.
- CS Table Friday: Heartbleed.
- Iowater project April 26 - Tag drains. Mail iowater@grinnell.edu for details.

Questions

Possible extensions to the project

Remember: You can extend both the parser and the unparser.

Remember: While I am talking about extensions in terms of difficulty, you are judged as much by the *utility* of the code.

Comparatively Easy

- Support whitespace in input.
- Error messages
 - Useful/helpful
 - Indicate precisely where the error is
 - Line/column
- Pretty printer

Moderate

- Pictorial error reporting
- Interactive error reporting
- Parse JSON into actual Java objects that match certain classes
`{"class":"Rational","numerator":1,"denominator":2} => new Rational(1,2)`
- Interactive object builder.
- Extend JSON.
- Support for unicode. `"\u1231"`

Difficult

- Output *any* object as JSON code.
- Parse JSON into actual Java objects that match the class.
`{"class":"Rational","numerator":1,"denominator":2} => new Rational(1,2)`
- Interactive object builder with audio GUI.

More notes on submitted work

- See document.

Planning presentations

- Monday.
- Email to csstudents goes out today. They get \$10 for showing up.
- Three minutes (no more than three slides)
- Send me no more than slides by Sunday at 8pm. One slide must have your names.
- Handouts optional.
- Prizes!
 - Gift cards!
 - Extra credit!
 - Maybe fruit snax.

Implementing heaps

- How do we know where the next element goes without looking through the tree or doing fancy math?
 - Similarly: How do we find the element to shove at the top when removing
- Basically: Finding the first space at the last level or the last element at the last level.
- Two ideas: Binary search trees (divide and conquer data structure) and hash tables (key/value and hashing; use an array for efficiency)
- Lets put the heap in an array using left-to-right, breadth-first, top-down order
- Shoving into an array.
- How do we find parent, left, and right

Heapsort

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.