CSC207.01 2014S, Class 35: Priority Queues and their Basic Implementation

Overview

- Preliminaries.
  - Admin.
  - Questions.
  - HW6.
- A few notes on priorities.
- Lab.

Preliminaries

Upcoming work.

- No writeup today!
- No reading for Friday.
- Exam due Friday!

Admin

- Pop Tarts!
- Consider running for the SEPC.
- Extra credit:
  - Convocation, today, noon: Elizabeth Kolbert (the Sixth Extinction)
  - Technology in the liberal arts symposium on Thursday.
  - CS table Friday: The Performance Contract, which is highly relevant to this class.
  - Dance in Chicago this weekend.

Questions on the exam

*What do you mean by "You can treat the height of the skip list as constant"*

If your algorithm is O(height), it’s O(1). (If you think carefully, height is about log_2(n).

*How do we call the order comparator in problem 1?*

```c
int order = order.compare(val1, val2); if (order < 0) { // Code for "val1 precedes val2" } // (order < 0) else if (order == 0) { // Code for "the values are the same" } // (order == 0) else // (order > 0) { // Code for "val1 follows val2" } // (order > 0)
```
How do we get the last problem to work if we adding/removing elements cause iterators to fail?

Use set rather than add or remove.

A Summary of HW6

Implement queues with arrays, with wrapping.

Implement the "pretty output" parenthesis matching algorithm.

Implement an RPN calculator.

- Reverse of Scheme. Operands appear before the operations
  - To add 3 4, "3 4 +"
- Why?
  - Complete unambiguous; no order of operations
  - Incredibly easy to implement with stacks

A few notes on priorities

- For priority queues, we use comparators to determine what the highest priority element is.
- Hidden assumption that we almost never write as a precondition
  - Transitivity: if a < b and b < c then a < c
  - Reflexitivity: if a < b then b > a
  - Complete: Any two elements are comparable

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.