CSC207.01 2013F, Class 39: Priority Queues and their Basic Implementation

Overview

- Preliminaries.
  - Admin.
  - Questions.
- A short introduction to priority queues.
- Array-based implementations.
- Sorting with priority queues.
- Running times.
- Lab.

Preliminaries

Admin

- I will reserve time for questions on HW9.
- There is no reading for tomorrow. Tomorrow we start to design a new interface that we will explore for the next few weeks (up through Turkey break).
- I'm still looking for time to get grading done. Most of the coming weekend is reserved for grading (both classes).
- Upcoming extra credit opportunities:
  - Town Hall, Wednesday, November 13, noon or 7:30 p.m.
  - *Debate, Tomorrow, Unknown time*
  - Learning from Alumni, Thursday, 2:15: Atul Gupta, Trustee
  - Tech Friday thingy in Des Moines. (Yes, you can miss my class Friday if you go. Do the lab on your own.)
  - CS Extra, Thursday, 4:15: Hilary Mason '00
  - Evening chat with Hilary Mason '00, Thursday, 8-9 pm
  - CS Table, Next Friday, HCI
  - Innovation Class, Friday, 12:45, ARH 302, Hilary Mason '00
  - Career Connections with Hilary Mason '00, Friday, 4:15-6:30 (???)
  - Swim meet, Saturday, Some time
  - ISO Dinner, Sunday
  - Digital Commons talk Monday, November 19, 7:00 p.m. or so
  - "Data Sovereignty: The Challenge of Geolocating Data in the Cloud", November 25, 4:15 JRC 101
  - "Gold Fever" by Andrew Sherburne '01 or so, 7:00 p.m., Monday, November 25, ARH 302

- Tuesday, November 20, 4:15 p.m., JRC 209 a gaming event with the game [d0x3d!]
- Unknown Startup Thingy in Des Moines

Questions on HW9

- Simple experiments are acceptable for verification.
- What's an "interesting" operation?
  - Something arithmetic is fine.
  - But stack operations are cool, too. "Rotate the top elements on the stack."

A short introduction to priority queues

- Linear structures:
  - "Deal with elements one at a time."
  - Primary operations: put and get.
  - A policy informs what value is returned by get.
- Priority queue: Give the highest priority thing back
- Priority determined by comparator.

Array-based implementations

Running times

- Unordered array
  - O(1) to add
  - O(n) to get
- Ordered array
  - O(n) to add
  - O(1) to get

Which is better?

- Depends on how you're using it.
  - n puts, 1 get - Use unsorted
  - n puts, n peeks, n gets - Use sorted
- Measure performance!
- Which is easier to implement?

Sorting with priority queues

- Step 1: Shove everything into the priority queue
- Step 2: Grab everything out of the priority queue

Can we do better?

- Stacks and queues: put and get are O(1)
- Priority queues: put or get is O(n)
- Can someone smart enough find an implementation of priority queues in which both put and get are O(1)
  - Yes: 2. Why?
  - No: those who read (and trust) Wikipedia
    - Argument: You've been told that there's no compare-and-swap algortihm that's faster than O(nlogn). Priority-queue-sort would be O(n) if both get and put are O(1)
- After break, we'll learn a really cool data structure that lets us do both put and get in O(logn). (For some stupid reason, this data structure is called a heap)