CSCA48 Summer 2017

Week 8 - Heaps

Brian Harrington

University of Toronto

June 19, 2017

UNIVERSITY OF TORONTO SCARBOROUGH

Admin

- I’m away at STLHE 2017 conference this week
- No lecture Wednesday
- Tutorials/Practicals as normal
- No office hours
- I’m still available via e-mail/Piazza

Priority Queues

- Like a Queue, but instead of always taking first, we want to take node with highest priority
- What this translates to: We want to take the largest/smallest first
- `enqueue(x, p)` - add a new element `x` with priority `p`
- `dequeue()` - remove and return the element with lowest priority score
- Usually also `is_empty()` and/or `size()`

Priority Queues

- How could we build this?
- Need to store in priority order, or be able to search by priority
    - Lists
    - Linked Lists
    - Binary Search Trees
- None of these are particularly helpful
- New data structure: Heaps!

HEAPS

- A Heap is a tree wherein all nodes obey the `heap property`
- The Min-Heap Property:
  - Let $p(x)$ be the value of x
  - For all nodes n:
  - For all children of n, $p(n) < p(child)$
- Max-Heap Property is exactly the same, but with $>$ instead of $<$
- A heap wich obeys the max-heap property is called a `max-heap`, if it obeys the min-heap property it's called a `min-heap`

HEAPS

- Some important facts about heaps:
    - The root contains the minimum (in the case of a min-heap) element in the tree
    - Every subtree of a heap is also a heap

HEAPS

- Let’s build one!