This thread is part of the proposed schedule for study for the first exam. Discuss the practice problems here.
View questions here: http://beastie.cs.ua.edu/concepts/cs/al/heaps.html
This thread covers questions 16-28 of heaps.

Work together on the proposed answers to questions on this shared Google Doc (comment reasoning/arguments behind answers)

Upcoming topics (threads start two days in advance): Graphs 1-15, Feb 15 | Graphs 16-28, Feb 16 | SB trees 26-52, Feb 17 | Recurrences 71-105, Feb 18
Full schedule

---

Subject: Re: Concept Review: Heaps (Part 2)
Posted by davidmccoy on Mon, 13 Feb 2017 17:07:22 GMT

I feel like both of these should be linear, because there’s no way to go about finding which subheap a value should be in quickly (unless keeping track of pointers to nodes).

---

Subject: Re: Concept Review: Heaps (Part 2)
Posted by lusth on Tue, 14 Feb 2017 13:51:59 GMT

That seems like a reasonable argument.

---

Subject: Re: Concept Review: Heaps (Part 2)
Posted by SSinischo on Thu, 16 Feb 2017 23:02:03 GMT

lusth wrote on Tue, 14 February 2017 07:51 That seems like a reasonable argument.

Why is it not log(n) time? The decreaseKey function takes log(n) time and involves finding the node to decrease.

EDIT: Figured it out; decreaseKey assumes you have a pointer to the node already.
Quote: Consider this set of operations: 15 inserts and one extraction of the minimum (in any order). What is the fewest / most number of subheaps found after the set is performed on an initially empty fibonacci heap?

I can't think of any cases where there is anything but 3 subheaps after consolidation.