CS240E W25	**Tutorial 2**	Jan. 17

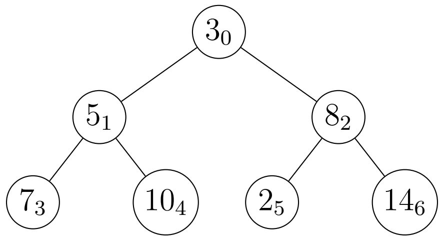
Overview

- Heapify
- Merging Binary Heaps
- Increase-key
- Binomial Heaps
- Converting between Binary Trees and Multiway Trees

Problems

Q1. Heapify.

Execute heapify on the following array (displayed here in tree form with array indices as subscripts):

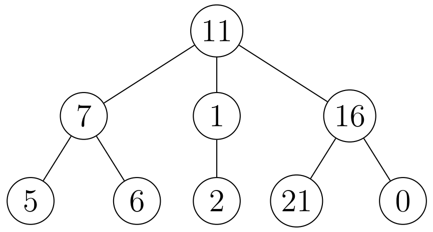


Q2. Merging Binary Heaps.

Let H_1 and H_2 be two binary heaps that both store exactly $2^h - 1$ items for some $h \in \mathbb{N}$. Show how to merge these two heaps. The run-time should be $O(h)$, presuming both heaps are stored as trees (not arrays), and the output must again be a binary heap (so satisfy the structural property).

Q3. Multi-way tree. Let T be a multi-way tree, i.e., nodes can have arbitrarily many children.

(a) Recall from lecture that there is a simple way to convert a multi-way tree T into a binary tree T': Each node of T also becomes a node in T', its leftmost child in T becomes the left child in T', and its sibling to the right in T becomes the right child in T'. Show the binary tree that you get in this way if you start with the following multi-way tree:

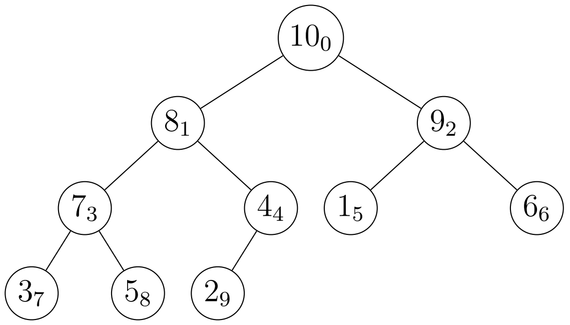


(b) For which binary trees T' is there a multiway tree T that it corresponds to? Justify your answer by explaining how you would convert such a binary tree T' into a multiway tree T.

(c) Assume T' is a flagged tree that satisfies the order-property of binomial heaps. What order-property does the corresponding multiway tree T have?

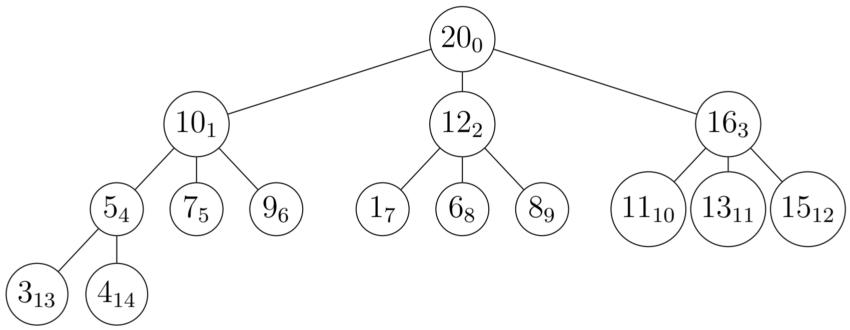
Q4. Increase-key.

(a) Here is a binary heap. Show the intermediate steps when performing `increase-key` $(8, 12)$ using fix-up.



(Recall that the first parameter of `increase-key` is a reference to a node, so in this case `increase-key` $(8, 12)$ means increase the key of the node at index 8 to 12.)

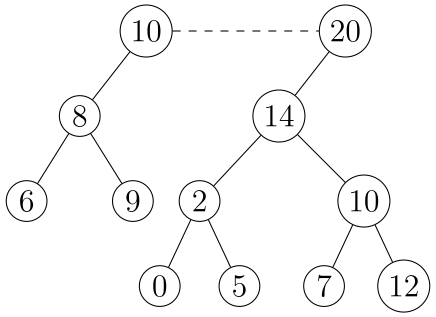
(b) Here is a multiway tree with the heap-order property. Show the intermediate steps when performing `increase-key` $(14, 19)$ using fix-up.



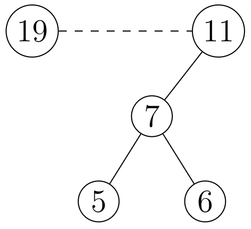
(Recall from lecture that we can't usually store multiway trees in arrays because nodes can have any number of children. But if we restrict this to each node having at most three children, then we can store it in an array, as shown by the indices in this example. Bonus question: How would you change the binary heap algorithms presented in class to work with this kind of tree structure?)

(c) What is the binary tree that corresponds to the multiway tree from (b)?

Q5. Binomial heaps. Perform the following operations on the binomial heap below, in order:



- Call `insert(4)`.
- Call `merge` with the following binomial heap:



- Call `deleteMax()`.