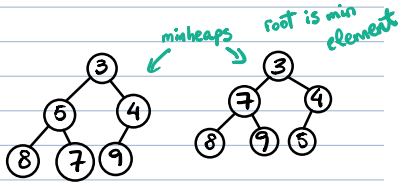
What is a Heap?

- It's a data structure to implement a priority queue.
- It's a complete binary tree in which each element is less than or equal to both of its children.
- It has both structural and ordering constraints
- As with search trees, there are many possible heap configurations for a given set of elements
- minheap, maxheap?

Complete: each level is filled



Operations on a heap

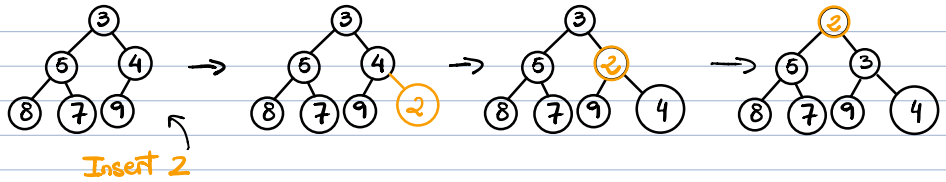
* addElement -> adds the given element to the heap

No sense to have other operations cause of constraints.

↳ Add the element as a leaf, keeping the tree complete.

↳ Then, move the element up toward the root, exchanging positions with its parent, until the relationship among the elements is appropriate

↳ This will guarantee that the resulting tree will conform to the heap criteria.



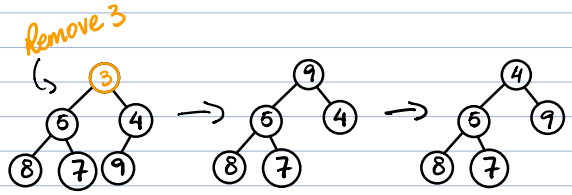
* removeMin -> removes the minimum element in the heap

↳ Remove the root (min) and reconstruct the heap

↳ First, move the last leaf of the tree to be the new root of the tree.

↳ Then, move it down the tree as needed until the relationships among the elements is appropriate.

↳ The algorithm is: compare the parent with its children and swap with the smallest if the child is less than the parent



* findMin -> returns a reference to the minimum element in the heap. -> Is it the root?

Implementing Heaps with links

- The operations on a heap require moving up the heap as well as down
- So we'll add a parent pointer to the HeapNode class, which is itself based on the node for a binary tree.
- In the heap itself, we'll keep track of a pointer so that we always know where the last leaf is.

Implementing Heaps with Arrays

→ With non-complete trees, there would be too much space wasted in the array.

- Since a heap is a complete tree, an array-based implementation is a good choice because it is very memory efficient
- Children stored at: $2n+1$ and $2n+2$ (parent element at index n)
- Parent of the node: $(n-1)/2$ → except the root

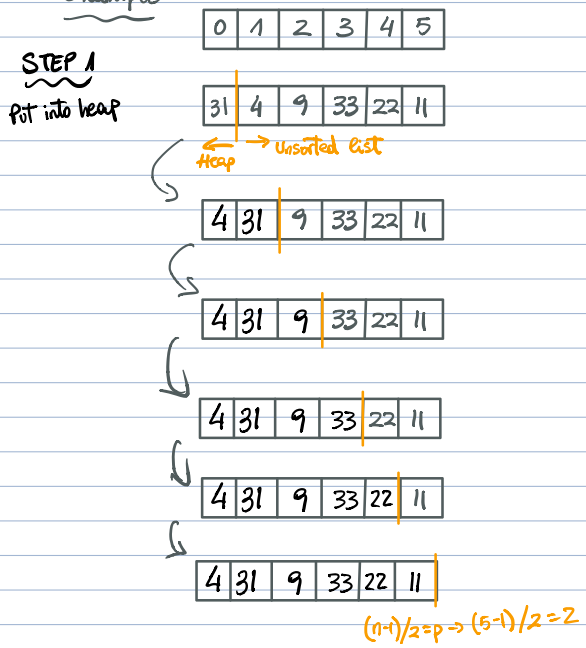
Heap Sort

- Good to use a heap to sort a list of numbers → because of ordering property.
- A heap sort sorts a set of elements by adding each one to a heap, then removing them one at a time
- The smallest element comes off the heap first, so the sequence will be ascending order.
- The array can be sorted "in place" no extra memory is needed.

Insert: $O(n \log n)$
Remove: $O(n \log n)$

Example:

STEP 1
Put into heap

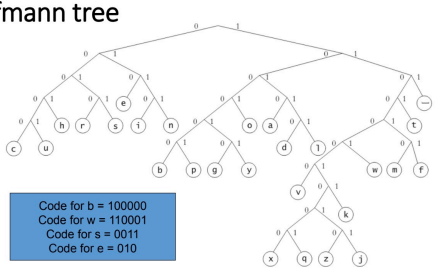


$(n-1)/2 = p \rightarrow (5-1)/2 = 2$

STEP 2
Sort heap

HOFFMAN TREE

Hoffmann tree



- The idea with a Hoffman tree is that many times we want to compress data and if I have a text (non-English capital letters, spaces) or some other kind of dataset, then it would a waste of memory to use ASCII.
- We create the Hoffman tree for our data, when we get a list of numbers (binary), we follow the path depending on if it is '0' or '1'.

110001000101...

Example:

* Binary string for "scissors cuts paper":

0011 00000 0110 0011 0011 1001 0010 0011 00000 00001 1101 0011 100001 1010 100001 010 0010
s c i s s o r s c u t + s P a P e r