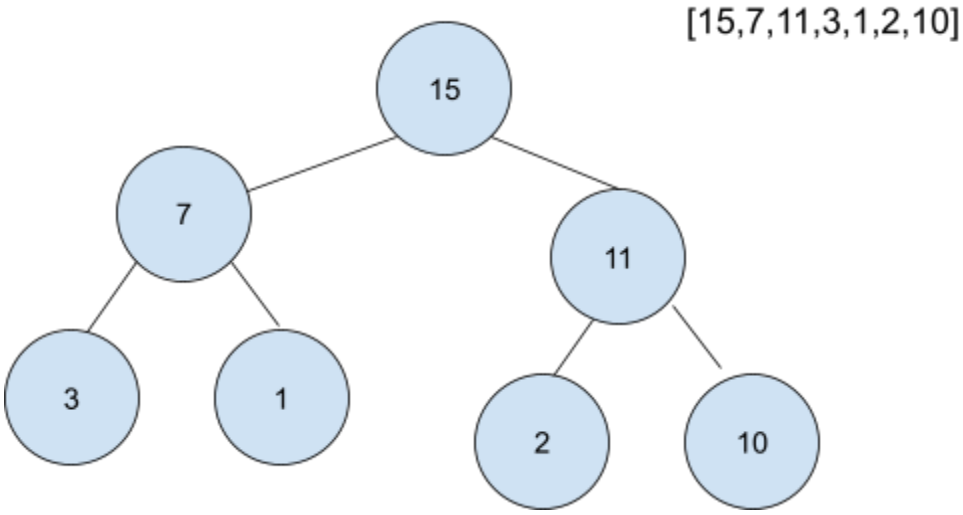
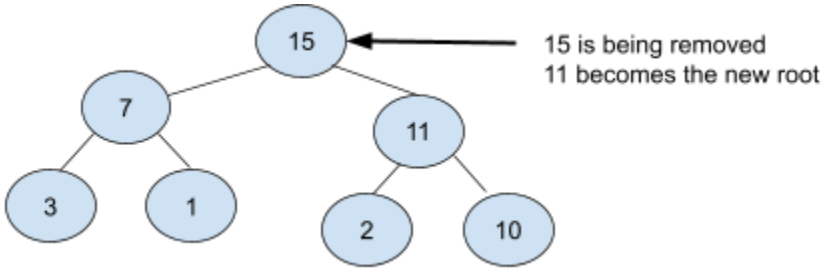
Josh Press
Professor Das
13 May 2024
Assignment 3:Implementation of a Binary Heap

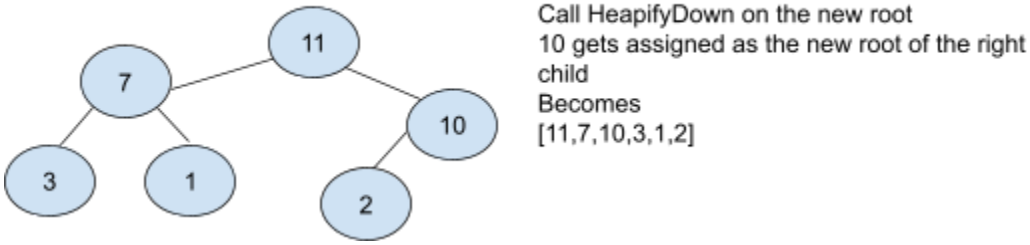
Introduction
The Binary heap data structure is important, especially for priority queues and extracting a maximum value quickly. For this assignment, I chose to implement a Binary Max Heap using arrays in Java since I was more comfortable with the array operations compared to a linked list.



For the operations, I focused on maintaining the heap's properties through recursion.
Methods

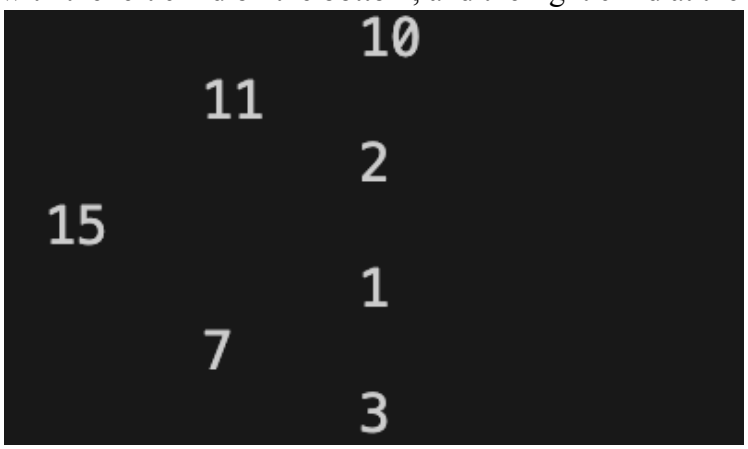
Insert()
For the insert method, the heap is checked to make sure that it is not full, then the value is placed at the end of the heap. heapifyUp is called on the last element to maintain the heap's properties.
HeapifyUp()
HeapifyUp that takes in a value, and gets the parent values. The value is compared to the other parent values until a place is found for the value to go into the heap. When a place is found for the value, the parent values swap.
Extact_Max()
Extract_Max works by removing the first element in the array (the max value), then setting the next value as the root and returning the value. Then HeapifyDown is called on the new root to adjust to the new value.
Example with [15,7,11,3,1,2,10]:





HeapifyDown()
HeapifyDown works by taking in a value getting the right and left children values. For my implementation, the array started at 1, so the left child is calculated by multiplying the index by 2. The right child is calculated by multiplying the index by 2, then adding 1.The left and right children values are compared, then swapped if needed to maintain the heap properties. If none of the children are smaller, HeapifyDown is called again.

Display()
My display method is similar to the one we implemented in class for the binary search tree. The array is printed, then the display recursive is called with the root.
Display_Recursive()
Display_recursive takes in an index and level then finds the children of the node. The parent node is printed, then Display_recursive is called on the left and right children, and the level is incremented until the heap is fully printed. The tree is printed horizontally in the command line, with the left child on the bottom, and the right child at the top.



Testing Process

For the testing process, I printed the array to make sure the heap properties were maintained since I worked on the Display method later. I tried adjusting the capacity and added different amounts of numbers to make sure my methods worked properly.

Challenges

One challenge I faced was when displaying the heap after extracting the max, the same value at the end would be printed twice. I resolved the issue by fixing my conditions to recurse and print the nodes.

Another challenge I faced was getting the display method to look similar to the one we implemented in class, since the heap would sometimes be printed without the correct spacing. I resolved this by thinking about the order of how Display_Recursive was called on the children.