

Heap Sort Explanation

In order to explain what a Heap Sort is, we need to explain what a binary search tree and a heap is. A binary search tree is a data structure that maintains nodes (collection of data) in a structured manner. In figure 1 you can see that child node on the left is less than the parent node and the child node on the right is greater than the parent node.

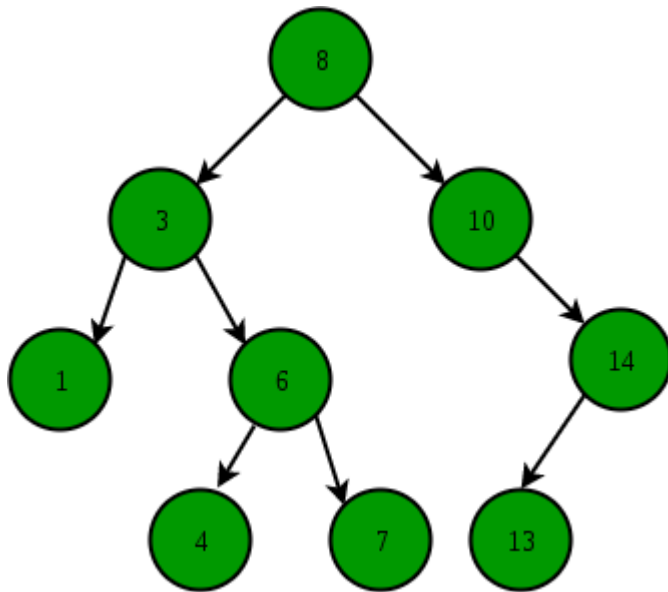


Figure 1

A heap is a completed binary tree where all of the levels, except the leaf node (last nodes), are completely filled. As shown in figure 2, the heap isn't in ordered and all of the parent nodes have two child nodes. The reason for this is because it's faster to find the smallest or biggest number in the tree than sorting the array.

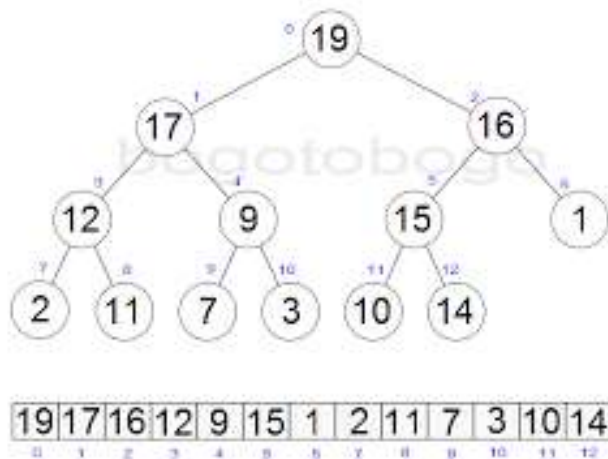


Figure 2

Heap Sort is an improved binary search tree because it doesn't have a quadratic worst case running time. It's divided into two parts. The first part is the creation of the heap using elements in an array. The second part involves repeatedly delete the root element in the heap formed in 1st place. The heaps can be sorted either into a max heap, where the root node can be greater than or equal to the child node and a min heap is when the node can be less than or equal to the child node.

Figures 4 to 7 show how a max heap is formed. Figure 4 is an unsorted array and figure 5 shows 89 being the biggest element in the array. It's then deleted and put to the back of the array. Figure 6 shows 81 being the second biggest, it's deleted and pushed down the unsorted array. This process repeats itself until it looks like figure 7, which is an example of how a max heap sort is formed.

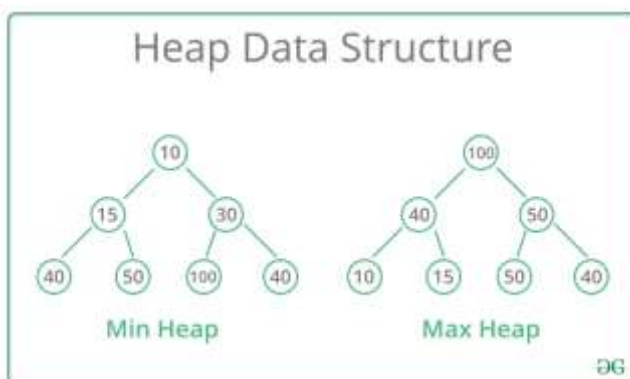


Figure 3

89	81	76	22	14	9	54	11
----	----	----	----	----	---	----	----

Figure 4

81	22	76	11	14	9	54	89
----	----	----	----	----	---	----	----

Figure 5

76	22	54	11	14	9	81	89
----	----	----	----	----	---	----	----

Figure 6

9	11	14	22	54	76	81	89
---	----	----	----	----	----	----	----

Figure 7

Sources:

Info:

<https://www.javatpoint.com/heap-sort>

<https://www.geeksforgeeks.org/heap-sort>

<https://www.geeksforgeeks.org/difference-between-min-heap-and-max-heap>

<https://www.geeksforgeeks.org/difference-between-binary-search-tree-and-binary-heap>

<https://www.baeldung.com/cs/heap-vs-binary-search-tree>

image:

<https://www.javatpoint.com/heap-sort>

https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.bogotobogo.com%2FAlgorithms%2Fimages%2Fheapsort%2Fheapsort_diagramA.png&imgrefurl=https%3A%2F%2Fwww.bogotobogo.com%2FAlgorithms%2Fheapsort.php&tbid=XCyLWhIJauGunM&vet=12ahUKEwiEiqn3pN_2AhU2YPEDHboCAT4QMygOegUIARD5AQ..i&docid=b_ImUJpTTfKfAM&w=715&h=564&q=heap%20sort&ved=2ahUKEwiEiqn3pN_2AhU2YPEDHboCAT4QMygOegUIARD5AQ

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fbinary-search-tree-data-structure%2F&psig=AOvVaw1h9JLqqevLxtCGAy0JANMY&ust=1648230020799000&source=images&cd=vfe&ved=0CAgQjRxqFwoTCMCxk8aI3_YCFQAAAAAdA AAAABAD