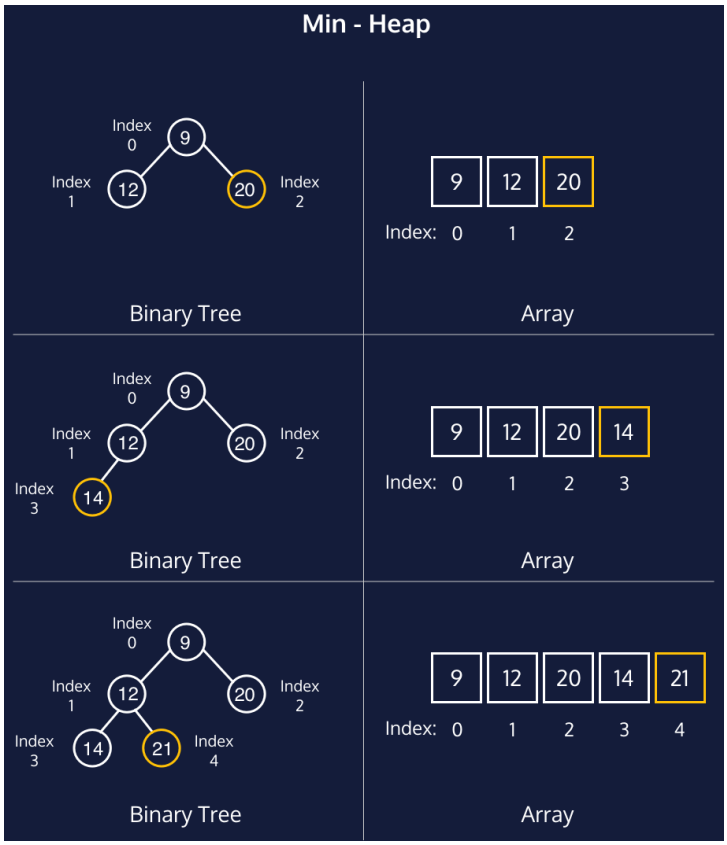
codecademy

Heaps

Heap Implementation

Heaps are typically implemented with a data structure such as an array or Python list. These sequential structures allow access to elements in a particular order which is key to efficient use of heaps. Although binary trees are helpful for understanding the relationships between nodes of a heap, implementation using a tree is less efficient for storage and retrieval of elements.



Adding Elements: Heapify Up

When a new element is added to a heap, if heap properties are violated, the new child must swap with its parent until both child and root properties are restored. This process is called *heapifying up*, because of the upwards movement of the new element in this restorative process.

Heaps as Binary Trees

A proper representation of a heap is a *complete binary tree*, which is a tree whose nodes have at most two children, and whose levels are filled completely from left to right (with no gaps in children). It's possible for the last level to be semi-completely filled, but all of its nodes are as far left as possible.