

Heap Representations

3 min

We can picture min-heaps as

Preview: Docs Loading link description

[binary](#)

trees, where each node has **at most** two children. As we add elements to the heap, they're added from left to right until we've filled the entire level.

At the top, we've filled the level containing 12 and 20. The next addition comes as the left child of 12, starting a new level in the tree. We would continue filling this level from left to right until 20 had its right child filled.

Conceptually, the tree representation is beneficial for understanding. Practically, we implement heaps in a sequential data structure like an

Preview: Docs Loading link description

[array](#)

or list for efficiency.

Notice how by filling the tree from left to right; we're leaving no gaps in the array. The location of each child or parent derives from a formula using the

Preview: Docs Loading link description

[index](#)

.

- left child: $(\text{index} * 2) + 1$
- right child: $(\text{index} * 2) + 2$
- parent: $(\text{index} - 1) / 2$ — **not used on the root!**

Instructions

Using the diagram, what child would we need to assign its own children before starting another level?

What would be the index of that new child on the next level?

Min - Heap



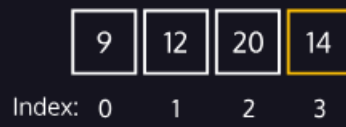
Binary Tree



Array



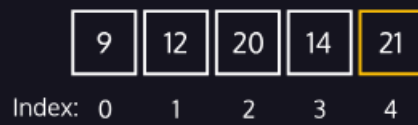
Binary Tree



Array



Binary Tree



Array