as we are tracking the execution of the algorithm

This is the reason why we often draw an array

array as its underlying data structure

since a binary heap is implemented using an array

(Insert, delete and sort) manipulate an array

In our lecture notes, all binary heap algorithms

Heap Algorithm Execution

A Note About the Tracking of Binary
memory space for the tree binary heap; hence it does not require additional.

However, understand that the algorithm does not create the tree representation of the heap when the binary heap is represented as a tree.

The reason we do this is because it is often easier to visualize how the algorithm executes.

However, as we are tracing the execution of the algorithm, we also often draw the tree representation of the binary heap.

Heap Algorithm Execution

A Note About the Tracing of Binary
Let's Try!

- Insert 5, 3, 2, 6, 0 into a Min binary heap

**Tree representation:**

**Array:**

```plaintext
elementCount
0 1 2
3 5 3 2
3 3
5
```
Min Binary Heap

Result: 02345

Array: 0 2 3 4 5 6

In heap: 0 1 2 3 4 5

In array: 0 2 3 4 5

Tree Representation:

Root: 0

Index: 1

Elements: 2 3 4 5 6

Node 0: Parent 1, Right 2
Node 1: Parent 0, Left 2, Right 3
Node 2: Parent 1, Left 3, Right 4
Node 3: Parent 2, Left 4, Right 5
Node 4: Parent 3, Left 5
Node 5: Right 6

Element Count:

Element Count: 5

In heap: 0 1 2 3 4 5

In array: 0 2 3 4 5

Tree Representation:

Root: 0

Index: 1

Elements: 2 3 4 5 6

Node 0: Parent 1, Right 2
Node 1: Parent 0, Left 2, Right 3
Node 2: Parent 1, Left 3, Right 4
Node 3: Parent 2, Left 4, Right 5
Node 4: Parent 3, Left 5, Right 6
Node 5: Right 6

Element Count:

Element Count: 5

In heap: 0 1 2 3 4 5

In array: 0 2 3 4 5

Tree Representation: