BS(start, end): middle = (start + end) / 2

**Base case 1:** array[middle] == value, return it
**Base case 2:** start > end, return nothing found

value < array[middle]: BS(start, middle - 1)
value > array[middle]: BS(middle + 1, end)

**Recursion:**
- 47
  - BS(0, 11)
  - BS(0, 4)
  - BS(6, 11)
  - BS(28)
    - BS(0, 1)
    - BS(3, 4)
    - BS(6, 7)
    - BS(9, 11)
    - BS(13)
      - BS(0, -1)
      - BS(1, 1)
      - BS(3, 2)
      - BS(6, 5)
      - BS(7, 7)
      - BS(9, 9)
      - BS(17)
      - BS(47)
      - BS(32)
      - BS(51)
      - BS(54)
      - BS(75)
      - BS(89)
      - BS(96)
      - BS(98)

**Array:**

0 1 2 3 4 5 6 7 8 9 10 11
13 17 28 32 33 47 51 54 75 89 96 98

**Tree Terms:**
- root
- leaves
- branch/subtree
- depth
- parent
- child
- sibling
- height
- degree
- size

**Balanced vs Unbalanced:**

For balanced binary trees:

\[
\text{degree}^{\text{height}} - 1 = \text{size}
\]

\[
\log(\text{size} + 1) = \text{height}
\]

\[
\text{degree is base of log}
\]

Worst case / least balanced:

\[
\text{height} = \text{size}
\]

Stepping through a balanced tree (top to bottom): \( O(\log(n)) \)
Stepping through an unbalanced tree (top to bottom): \( O(n) \)