Step-by-step Logic

1. **Initialize pointers:**
    - `left = 0`
    - `right = n - 1`
2. **Loop while `left < right`:**
    - Find `mid = Math.floor((left + right) / 2)`
3. **Compare `arr[mid]` with `arr[right]`:**
    - **Case 1:** `arr[mid] > arr[right]`
        - This means the smallest element is **to the right of mid**.
        - So, set `left = mid + 1`.
    - **Case 2:** `arr[mid] <= arr[right]`
        - This means the smallest element is **at mid or to the left of mid**.
        - So, set `right = mid`.
4. **Loop ends when `left == right`**
    - Both will point to the smallest element.
5. **Return `left` (or `right`) as the pivot index.**

Example Walkthrough

For `[4, 5, 6, 7, 0, 1, 2]`:

- Initial: `left=0`, `right=6`
- 1st Iteration: `mid=3` (`arr[3]=7`, `arr[6]=2`)
    - `7 > 2` ⇒ move `left` to `mid+1 = 4`
- 2nd Iteration: `left=4`, `right=6`, `mid=5`
    - `arr[5]=1`, `arr[6]=2` ⇒ `1 < 2` ⇒ move `right = mid = 5`
- 3rd Iteration: `left=4`, `right=5`, `mid=4`
    - `arr[4]=0`, `arr[5]=1` ⇒ `0 < 1` ⇒ move `right = mid = 4`
- Now `left=right=4`
    - Pivot is at index 4.