- **heapify:**
    - Ensures the Max-Heap property for a subtree rooted at index i.
    - Compares the root with its left and right children and swaps if the root is not the largest.
    - Recursively fixes the affected subtree if needed.
- **heap_sort:**
    - Builds a Max-Heap from the array.
    - Repeatedly swaps the root (largest element) with the last element, reduces the heap size, and rebuilds the heap.
    - Continues until the entire array is sorted.

Execution Steps:

1. Build a Max-Heap from the array.
2. Extract the largest element (root) and place it at the end of the array.
3. Reduce the heap size and rebuild the Max-Heap for the remaining elements.
4. Repeat until the array is sorted.

Example:

Input: [4, 1, 3, 9, 7]

- After building Max-Heap: [9, 7, 3, 1, 4]
- After sorting: [1, 3, 4, 7, 9]

Time Complexity:

- Building the heap: O(n).
- Sorting using the heap: O(nlogn).
- Total: O(nlogn).