1. **Introduction to Heap / Prority Queue :** A Powerful Simple Data Structure based on Binary Tree. Heap is a Complete Binary Tree.

    a. aInsert : O(log N)
    b. Max / Min : O (1)
    c. Delete : O(log N)

2. **Heap Insertion :** Theory
    a. Max heap for finding maximum value O(1),
    b. Min heap for finding minimum value O(1).
    c. For **Max** heap all child nodes are equal or **less** than their parent.
    d. For **Min** heap all child nodes are equal or **greater** than their parent.
    e. So Max / Min value will be on the root. That 's why O(1)

    Max Heap Insert : Up-heapify (compare with parent & swap) O()

3. **Heap Insertion :** Implementation
    O(Log N): [Height = Log N, Complete Binary Tree]
        A. Using Dynamic Array
        B. Left Child : i = 2i + 1
        C. Right Child : i = 2i - 1
        D. prrent = i-1/2

4. **Heap Delete :** Theory [For Max Heap]
    Swap with last node. then delete the last node O(1), Then call down-heapify
    Down-heapify : compare parent with its two children and swap downward.
    Complexity : O(Log N)

5. **Heap Delete :** Implementation
6. **Heap Max Operations** : Other Operations : getMax, ExtractMax().