1. **Build Heap from Array:** Converting an array to a heap. O(n)
   From last non-leaf node to root, we need to call down_heapify()
   Leaf node = the node who doesn't have any child.
   Last non-leaf node index = n/2 -1

2. **Build Heap from Array Complexity:** O(N) [Not O(n log n)], Explanation

   Each height have **ceil (n/ 2^{h+1})** nodes

3. **Heap Sort :** Complexity - O(n Log n)
   a. Convert array to heap
   b. ExtractMax(), push and delete
   c. Reverse, [if we use MinHeap we don't need to reverse]

4. **Priority Queue:** Based on Heap. Its a Non-Linear DS
   a. Push O(Log n) [Insert]
   b. Top(print max value) O(1) [root of heap]
   c. Pop(delete max value) O(Log n) [Delete]
   d. Size.

5. **STL Priority Queue :** Implementation
6. **Priority Queue Example Problem**