O (n\*2)

1. Nested for loop,
2. while
3. three pointers, outer pointer - O(n) with (left and right right pointer) O(n)- e.g three sum

O (n log n)

1. priority queue - n \* (log n)
2. Sorting
3. Sorting + binary search
4. Sorting Comparable objects

O (n)

1. One traversal + cache value
2. Two traversals - one potential - one validation
3. Two pointers
   1. slow and fast runners
   2. opposite ends (container with most water)
4. two pass - one from right, one from left (trapping rain water)
5. Shuffling
6. QuickSelect algorithm ( k’th element from right in non-sorted array)
7. Use pointers for 3 arrays. Move other pointers until or less than current pointer or until end of the array.

O (log n)

1. Binary search
2. Search in BST

Tree:

Level order: use q.size and poll untill size is 0, for calculating elements belonging to that level.