Algorithm

Sieve of Eratosthenes

MAIN IDEA | Cross out all the multiple number from $2 \rightarrow n$

EXAMPLE: Find all prime number in the following list

2, 3, 4, 5, 6, 7, 8, 9, 10

Step 1: 2, 3, \(\frac{1}{4}\), 5, \(\frac{1}{6}\), 7, \(\frac{1}{6}\), 9, \(\frac{1}{6}\)

Step 2: 2, 3, \(\), 5, \(\), 7, \(\), \(\), \(\)

Prime: 2, 3, 5, 7

Merge Sort

MAIN IDEA | Find the mid point and then split array into half

Merge sort the left array

Merge sort the right array

Merge left and right array

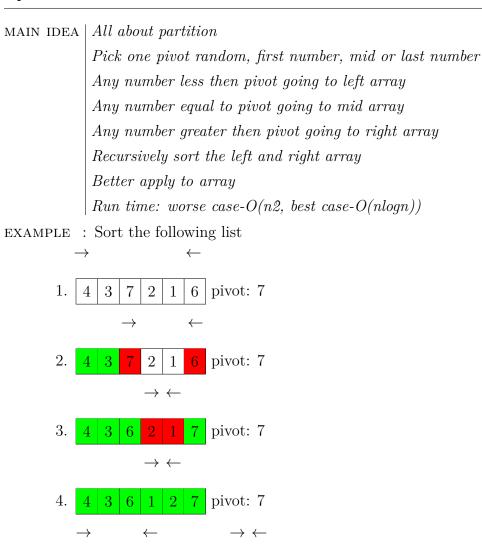
Better apply to LinkedList

Rune time: O(nlogn)

 ${\tt EXAMPLE}\;$: Sort the following list

- 1. 4 3 7 2 1 6
 - \checkmark
- 2. 4 3 7 2 1 6
- 3. 4 3 7 2 1 6
 - $\langle \ \rangle$ $\langle \ \rangle$
- 4. 4 3 7 2 1 6
 - \ \ \ \ \ \ \
- 5. 3 4 7 1 2 6
 - \searrow \swarrow \searrow \swarrow
- 6. 3 4 7 1 2 6
 - \searrow \swarrow
- 7. | 1 | 2 | 3 | 4 | 6 | 7

Quick Sort



2 | pivot: 3 | 1

pivot: 1

Quick Select

5.

MAIN IDEA | Similar as quick sort

2 | 3 | 5 | 6 | 7

6

The difference is quick sort recursively sort left, mid and right Quick Select only recursively sort left or right

Binary Search

 ${\tt MAIN\ IDEA} \mid \mathit{Find\ mid\ point\ then\ compare\ with\ the\ target}$

Base case if the mid is equal to target then return it

If the mid is less then target recursive left half

If the mid if greater than target recursive right half

Run time: O(logn)

EXAMPLE: Find 7 in the following sorted array

1 3 5 7 8

Μ \mathbf{R} L

 $1 \mid 3 \mid 5 \mid 7 \mid 8 \mid \text{mid: } 5, 5 < 7$

L M R

| 1 | 3 | 5 | 7 | 8 |mid: 7, 7 == 7