Array

Linear

Data Structure

n = 7

Index Number	0	1	2	3	4	5	6
arr	79	87	82	57	95	98	69
Base address	100	104	108	112	116	120	124

contiguous manner

storage space

int = 4 Bytes

0

1

1 Byte = 8 bits

arr[4] Index Number ——— Random access

→ 95

Static Array

fixed array size

↓

```
int[] arr = new
        int[10];
```

Dynamic Array

Change in the size of an array runtime

Java

ArrayList

Python

List

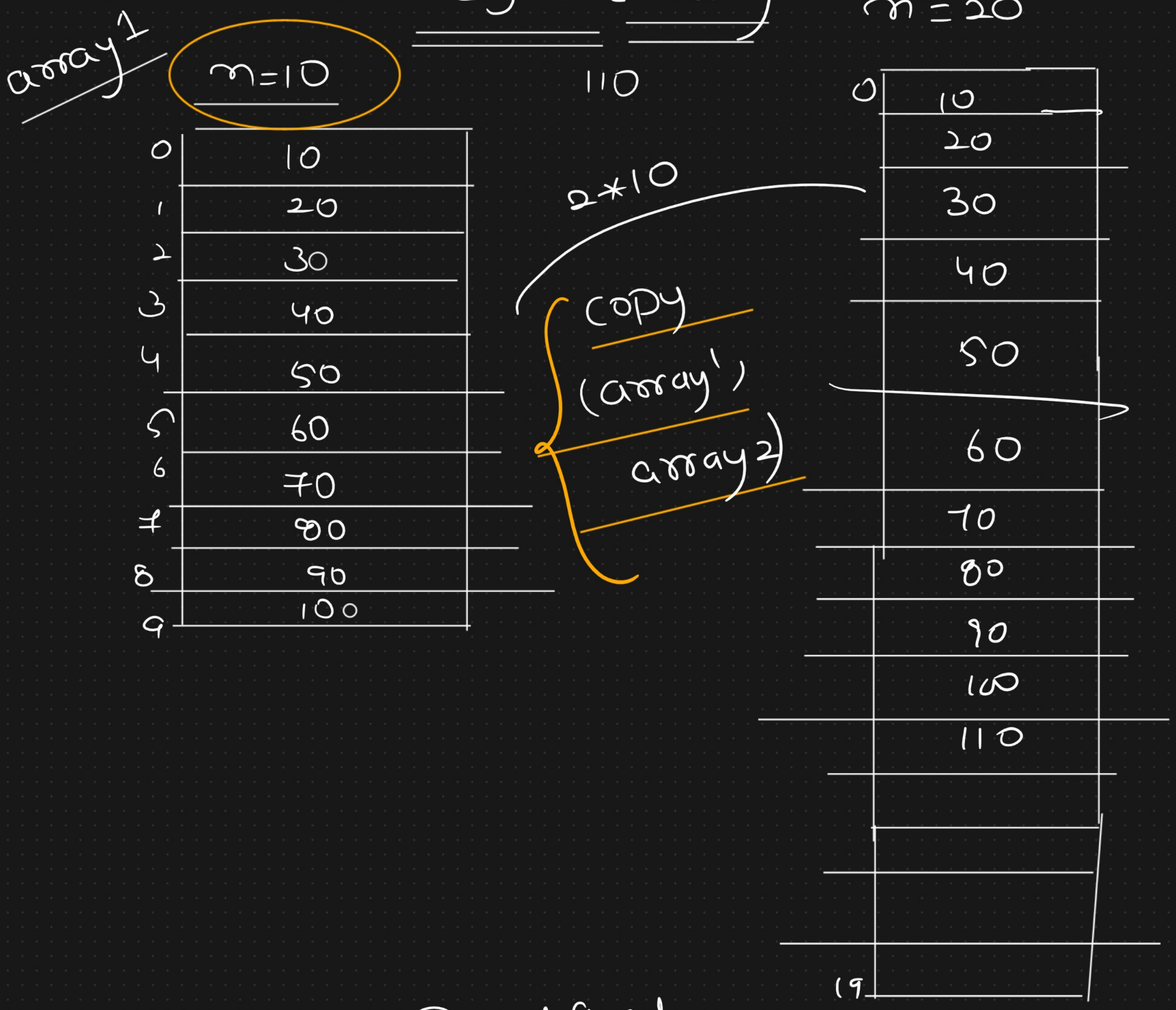
- Javascript
- C++
- C
- Python
- Ruby
- Java
- Mojo
- Pearl

Algorithm

↓

It is independent of any Programming Language

Dynamic Array



Pre-defined

sort

outplace

(

MergeSort/

QuickSort

$O(n\log n)$

Worst

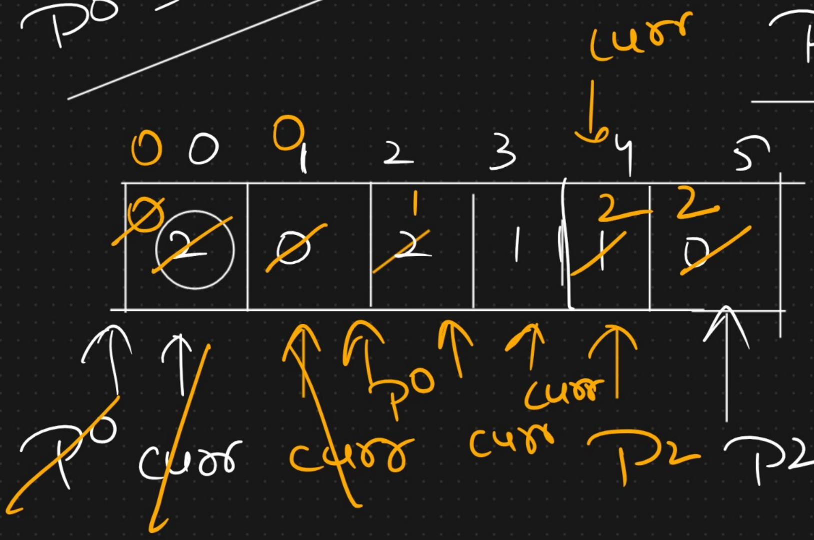
$O(n^2)$

Inplace

↳ We are not using any extra space outside the given array

P0 = curr = 0

P2 = 5

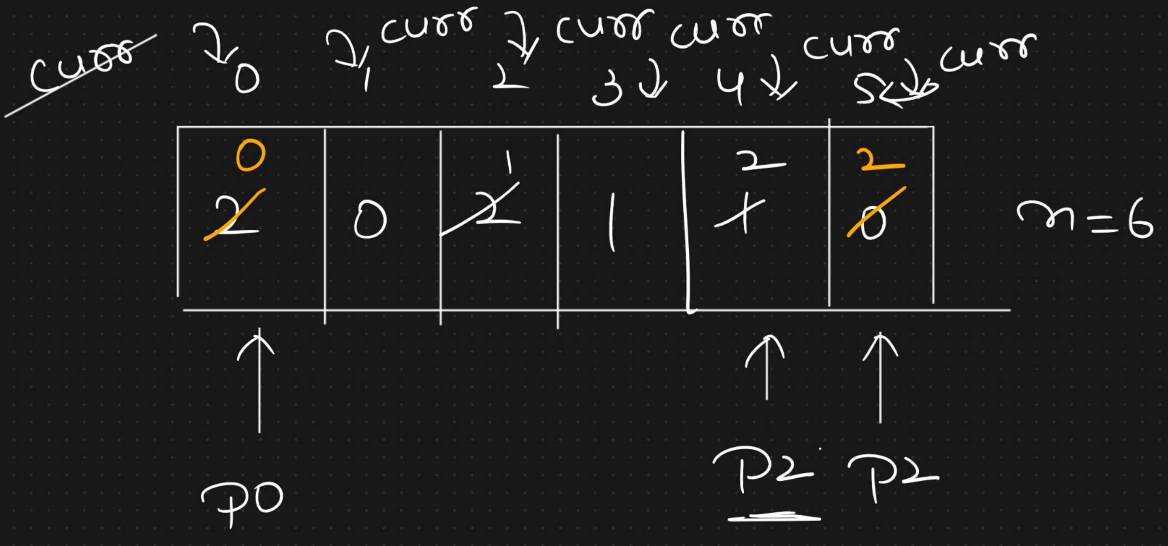


P0 — to take 0's

P2 — to take 2's

P2 — 2's

P0 — 0's

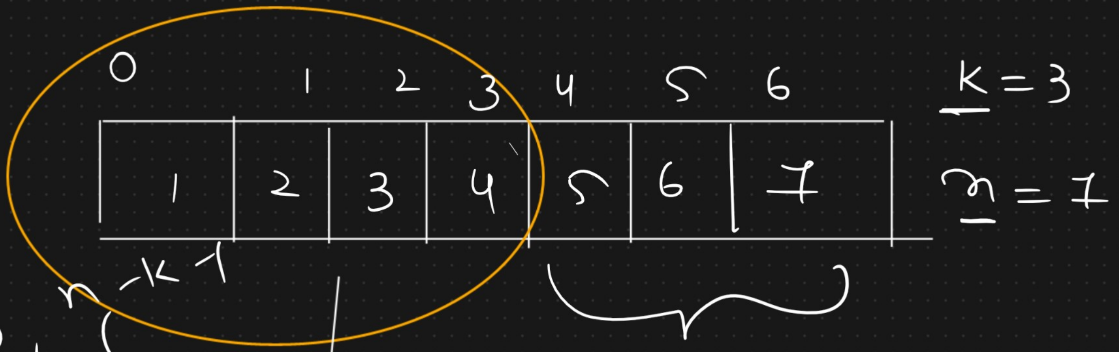


0, 0, 1, 1, 2, 2

Space complexity

↳ Extra space (Not using

↳ O(1)



0 l (

1) Reverse(0,3)

0 1 2 3 4 5 6

4, 3, 2, 1, 5, 6, 7

2) Reverse(4,6) n-1

n-k

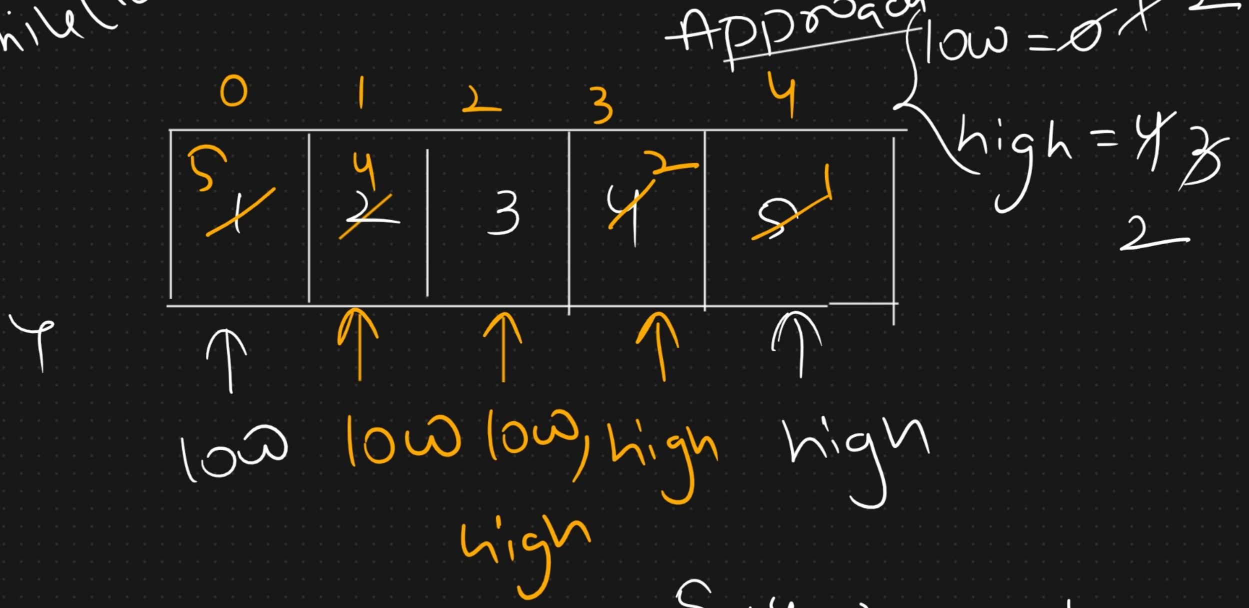
4, 3, 2, 1, 7, 6, 5

5, 6, 7, 1, 2, 3, 4

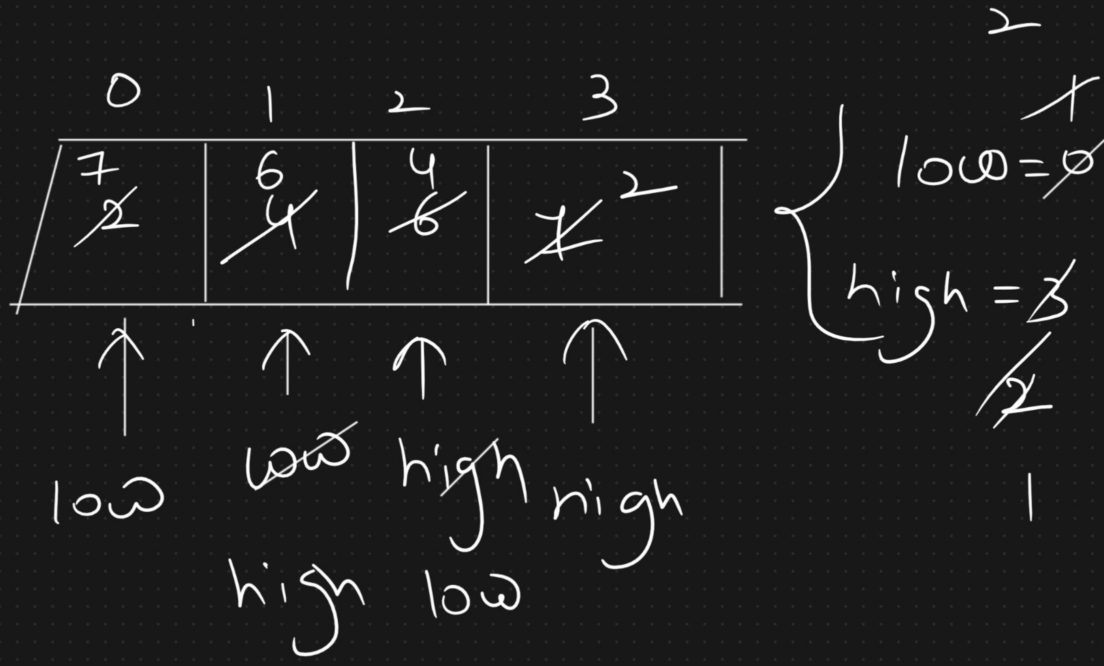
3) Reverse(0, 6)

While(low<high){

Two Pointer Approach



5, 4, 3, 2, 1



$K = K \% n$ Remainder

$= 13 \% 5$

$= 3$

$n = 5$

$K = 13$

$K = 3$	$K = 13$