Friday, February 24, 2023 7:51 PM

[10,9,2,5,3,7,101,18]

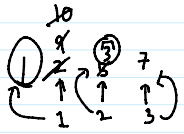
From <https://leetcode.com/problems/longest-increasing-subsequence/>

data ≥ ali

prev element

length

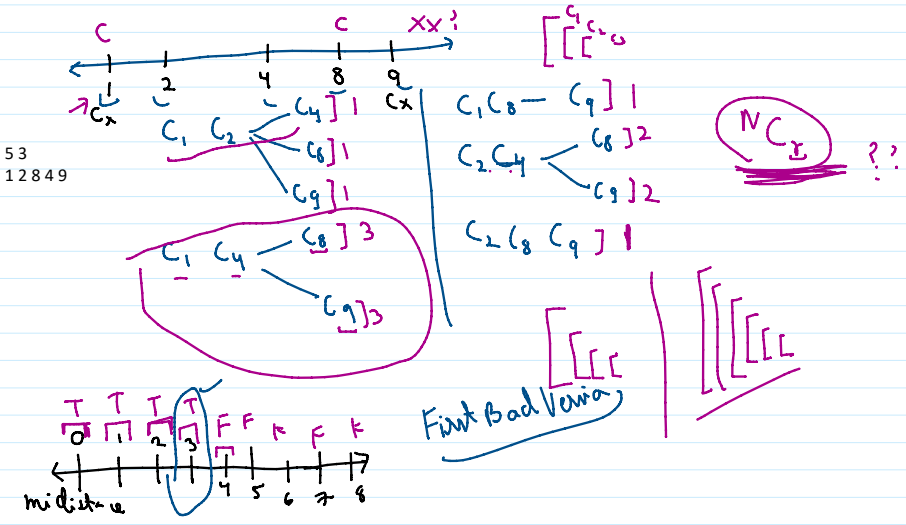
[10,9,2,5,3,7,1,2,101,18,8,3]

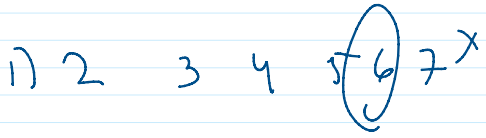


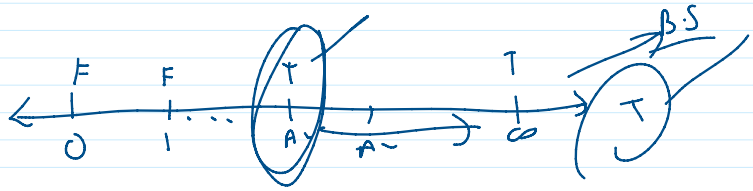
aababbb

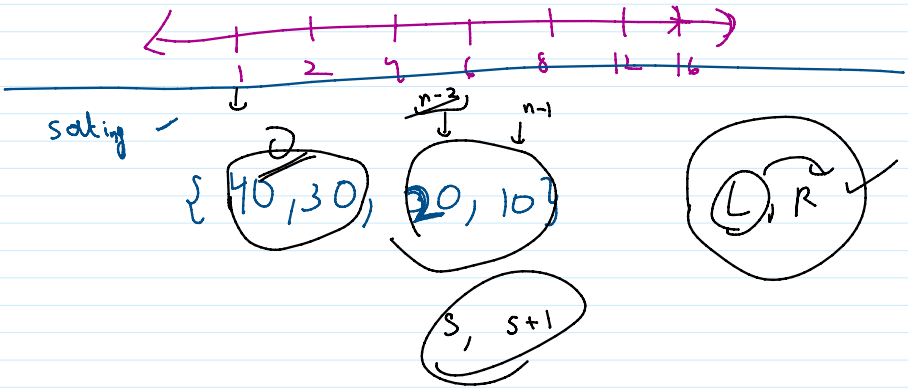
From <https://leetcode.com/problems/substring-with-largest-variance/>

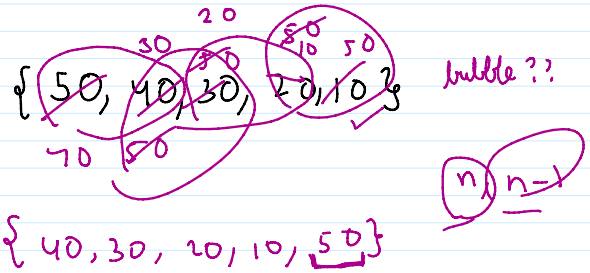
5












```java
public static void bubble(int[] arr) {
    for (int cnt = 1; cnt < arr.length; cnt++) {
        for (int s = 0; s <= arr.length - 2; s++) {
        s and s+1
        L and R
            if (arr[s] > arr[s + 1]) {
                int temp = arr[s];
                arr[s] = arr[s + 1];
                arr[s + 1] = temp;
            }
        }
    }
}
```

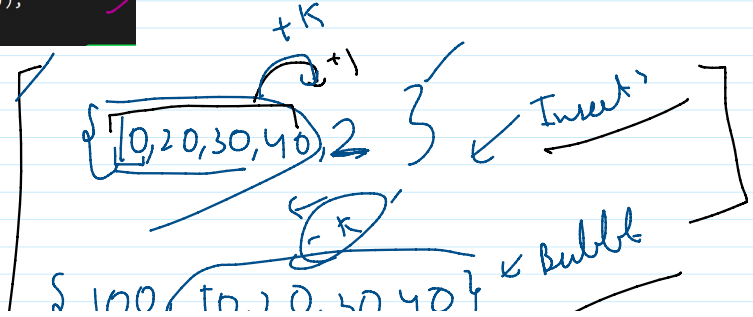
WS, BS

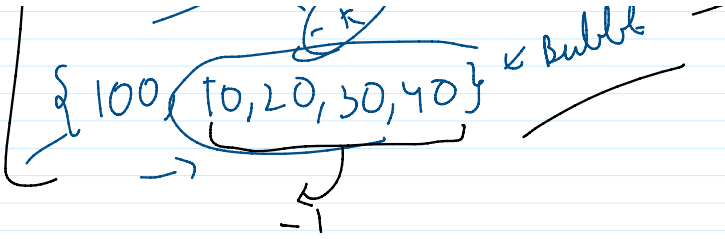
{50, 40, 30, 20}

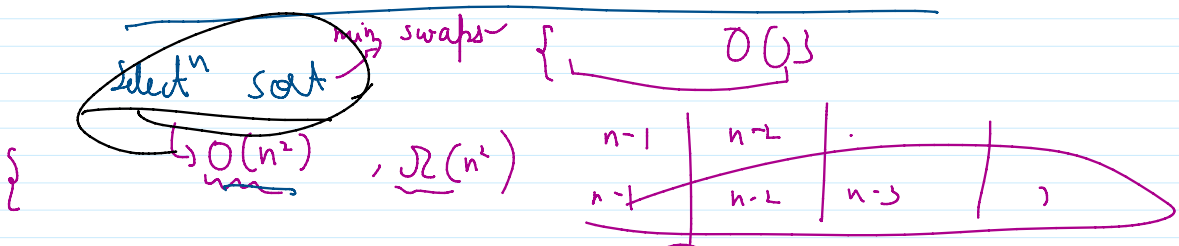
sort

Bubble → $O(n^2)$	$\Omega(n^2)$ / $\Omega(n)$
Insertn → $O(n^2)$	$\Omega(n)$

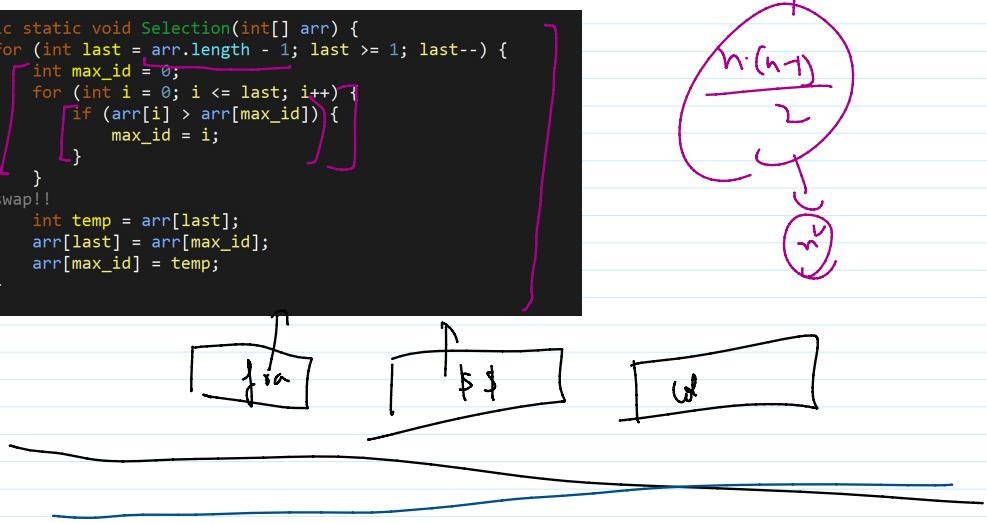
```java
public static void Insertion(int[] arr) {

    for (int to_ins = 1; to_ins < arr.length; to_ins++) {
        int idx = to_ins - 1;
        while (idx >= 0 && arr[idx] > arr[idx + 1]) {
            int temp = arr[idx];
            arr[idx] = arr[idx + 1];
            arr[idx + 1] = temp;
            idx--;
        }
        System.out.println(Arrays.toString(arr));
    }
}
```








```java
public static void Selection(int[] arr) {
    for (int last = arr.length - 1; last >= 1; last--) {
        int max_id = 0;
        for (int i = 0; i <= last; i++) {
            if (arr[i] > arr[max_id]) {
                max_id = i;
            }
        }
    swap!!
        int temp = arr[last];
        arr[last] = arr[max_id];
        arr[max_id] = temp;
    }
}
```

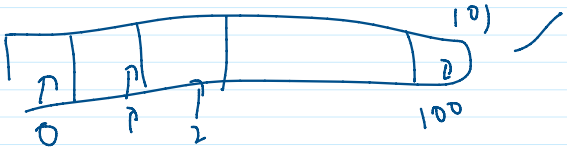


{ 0,0,1,0 , 1, 1, 0, 0, 1, 1, 0 . . }

{ 0, 1, 2, 2, 2, 1, 1, 2, 0, 1 } — 3

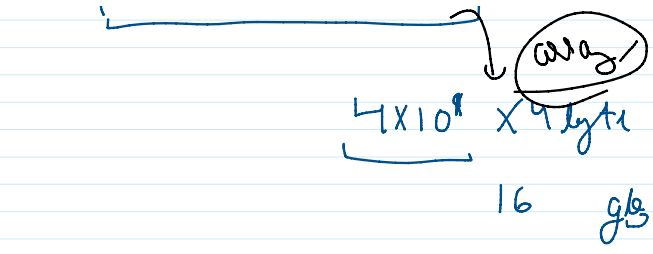
(count sort
Ω(n)
O(n)

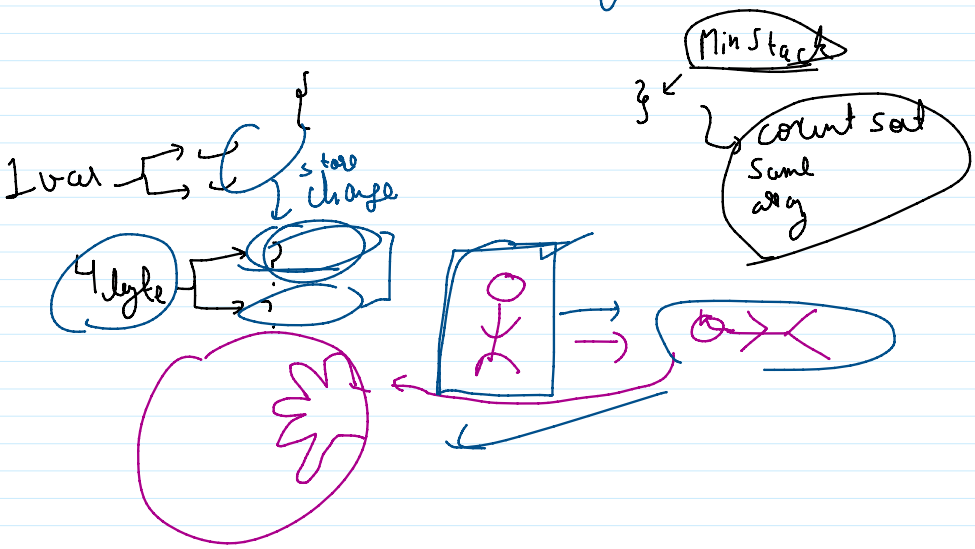
{ 0, 1 . . . . 100 }

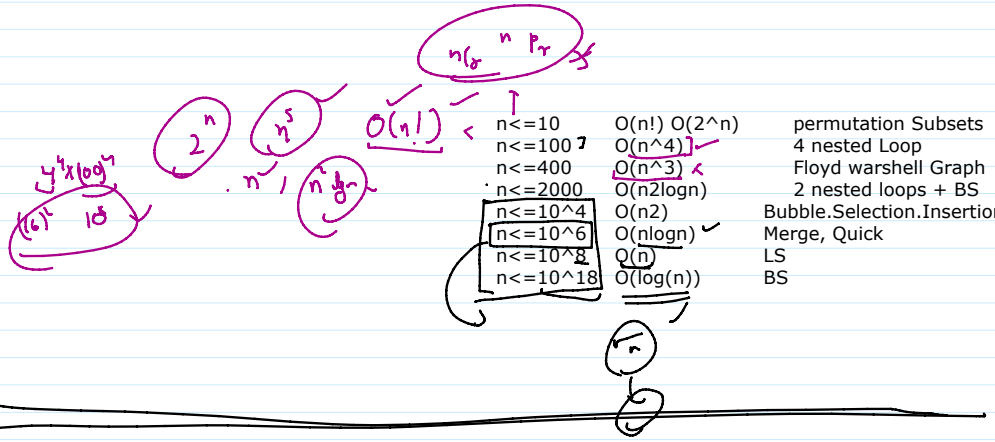


-2 x 10^9 to 2 x 10^9

avg

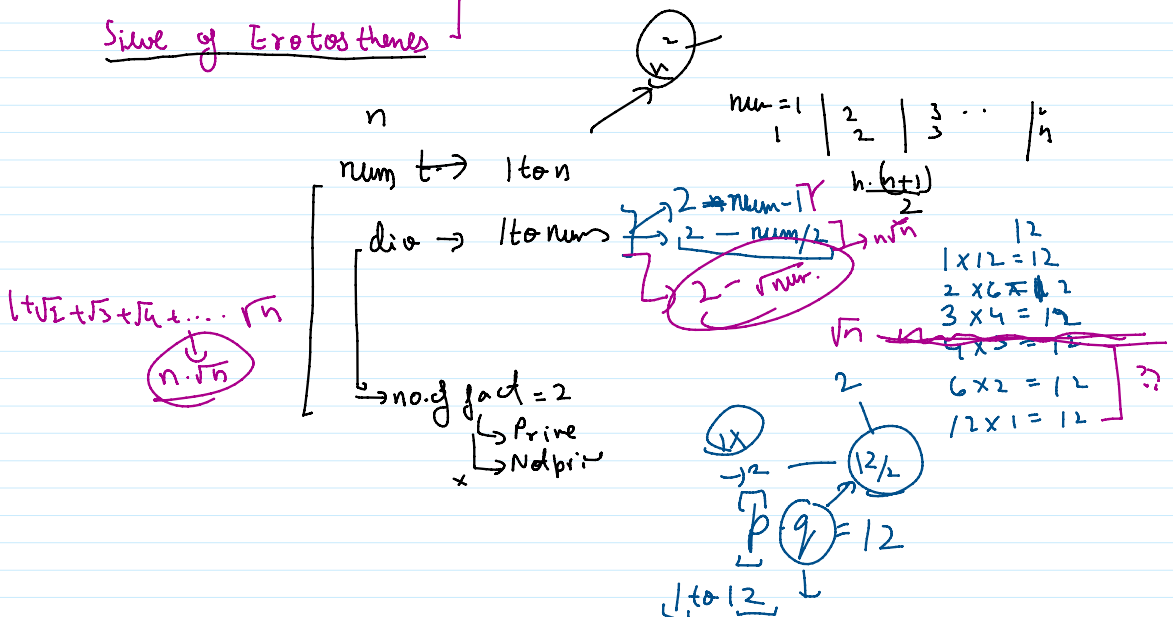






SOE

Sieve of Erotosthenes



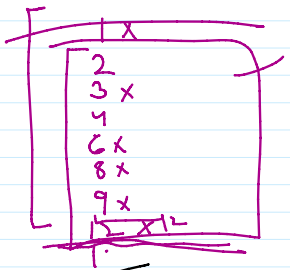
P(4) = 12

1 to 12

12 to 1

2

144



num → 1 to n

div → 1 to num

→ no. of fact = 2
→ Prime
→ Not prime

num = 2 → 2 to num-1 / 2 to 1

A

B

10

div = 1
1 to n

div = 2
2, 3, 4, 5, 6, ..., n → n-2 → n/2

div = 3
num = 3, 4, 5, 6, 7, ... n → n-3 → n/3

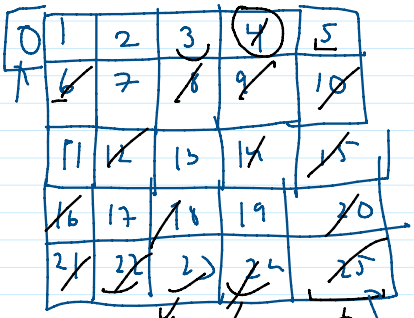
div = 4
num = 4, 5, 6, ... n → n-4 → n/4

div
n + n-1 + n-2 + ... 1

div
n + n/2 + n/3 + n/4 + n/5 + ... n/n

n(1 + 1/2 + 1/3 + ... 1/n) = n log n

n = 25



2 to √n

div = 1 to 25

div
2
3
4
x
5

7

2 to $\sqrt{25}$ ← T

div = 1 to 25

2 to 25/2

2 to $\sqrt{25}$

div = 2 to $\sqrt{num}$

div

```java
boolean[] isComposite = new boolean[n + 1];
for (int div = 2; div * div <= n; div++) {
    if (isComposite[div] == false) {
        for (int table = div * 2; table <= n; table = table + div) {
        System.out.println("composite="+table);
            isComposite[table] = true;
        }
    }
    System.out.println("==========");
}
for (int num = 2; num <= n; num++) {
    if (isComposite[num] == false) {
        System.out.println(num);
    }
}
```

$n/2 + n/3 + n/5 + n/7$

$n\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13}\right)$

$n \log(\log_2 n)$ → n

div = 2

2x2
2x3
2x4
2x5
:

div = 3

3x2
3x3
3x4
3x5

div = 4

4x2
4x3
4x4
4x5

div = 5

5x2
5x3
5x4
5x5
5x6

div = 6

6x2
6x3
6x4
6x5
6x6

div = 7

7x2
7x3

7x7