

$[0, pt] \rightarrow 0's$

$[pt, i+1] \rightarrow 1's$

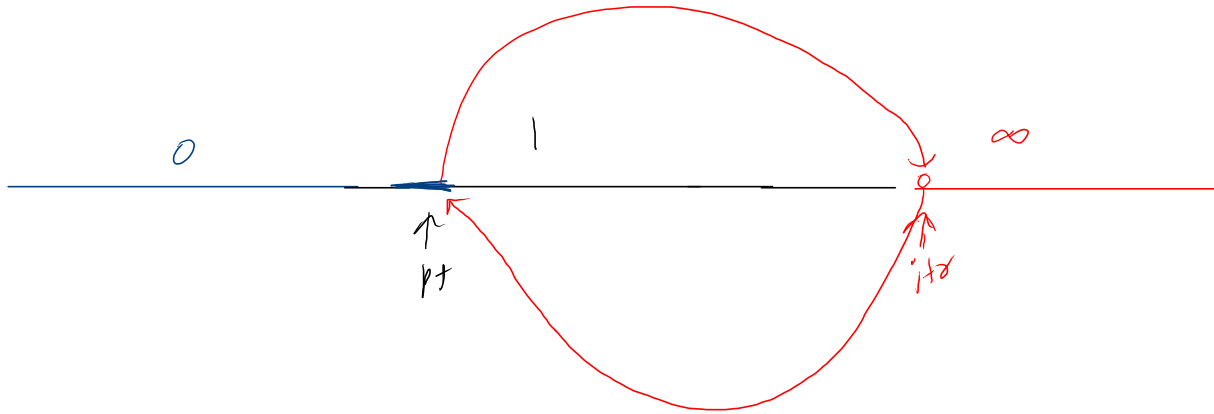
$[i+1, n] \rightarrow \infty$

array

0	1	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---

$T: O(n)$

$S: O(1)$



2	8	9	3	1	1	4	17
0	1	2	3	4	5	6	7

, index = 6 →

data = arr[index]

0	1	2	3	4	5	6	7
2	9	1	3	1	9	8	14

↑

⑦

$(0 \leq \text{index} \leq n-1)$

$(\text{element} \leq \text{data})$

$(\text{ele} > \text{data})$

$\text{ele} \leq \text{data}$

$\text{ele} > \text{data}$

$\infty$

2	3	1	7	2	3	④	14	8	9	19	20
0	1	2	3	4	5	6	7	8	9	10	11

pt  
0

9

index = 6

data = 9

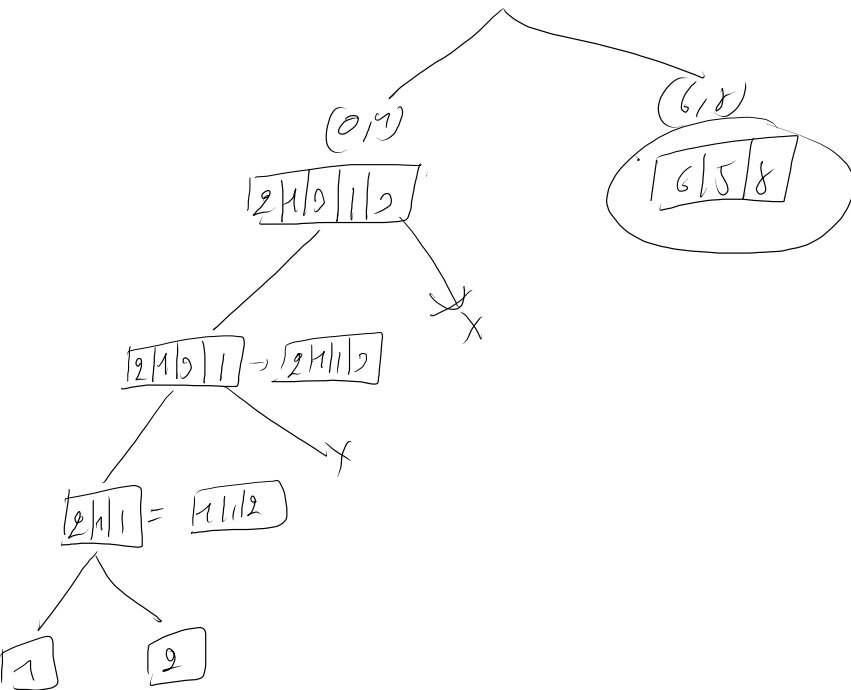
index = (n-1)

data = arr[x]

4	2	3	④	14	12	8	6	10	11	9
---	---	---	---	----	----	---	---	----	----	---

2	1	3	6	7	1	3	5	8
0	1	2	3	4	5	6	7	8

2	1	2	1	3	4	6	5	8
0	1	2	2	7	5	6	7	8



cost  $\rightarrow$  (mid is also very big str in array)

```

T(h)
def quickSort(arr, si, ei):
    if si >= ei:
        return

    mid = (si + ei) // 2
    pivotIdx = segregateElements_02(arr, mid) # O(N)

    quickSort(arr, si, pivotIdx - 1)
    quickSort(arr, pivotIdx + 1, ei)

```

Ans

$$T(h) = h + T(h/2) + T(h/2)$$

$$T(h) = h + 2T(h/2)$$

$$\downarrow$$

$$T(N) = N + N \log(N)$$

work

$$\begin{aligned}
 T(N) &= N + \cancel{T(N-1)} \\
 \cancel{T(N-1)} &= (N-1) + \cancel{T(N-2)} \\
 \cancel{T(N-2)} &= (N-2) + \cancel{T(N-3)} \\
 &\vdots \\
 \cancel{T(1)} &= 1 + \cancel{T(0)}
 \end{aligned}$$


---


$$T(N) = 1 + 2 + 2 + \dots + (N-2) + (N-1) + N$$

$$= \frac{(N)(N+1)}{2}$$

$$T(N) \approx O(N^2)$$

$$N \log_2(N) \rightarrow N, N/2, N/4, \dots$$

$$\boxed{N \log_2(N)} \rightarrow N, N/2, N/4, \frac{N}{8}, \frac{N}{16}, \frac{N}{32}, \dots$$