$[0, pt]$ → 0's

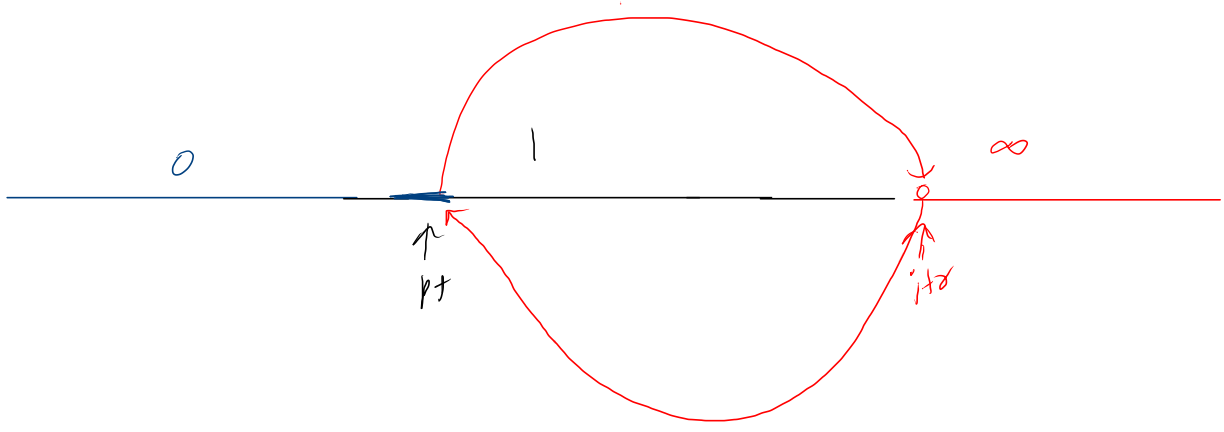
$[pt+1, itr-1]$ → 1's

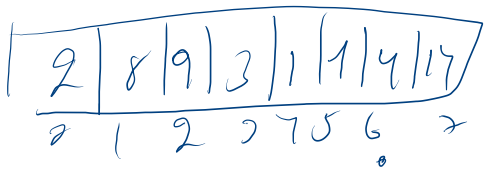
$[itr, n-1]$ → ∞

array → | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

T: O(n)

S: O(1)





, index = 6

data = arr[index]

$(0 \le index \le 4)$

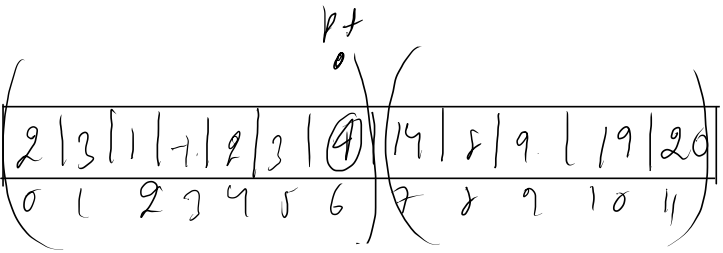
$(element \le data)$

$(ele > data)$

ele ≤ data

ele > data

∞



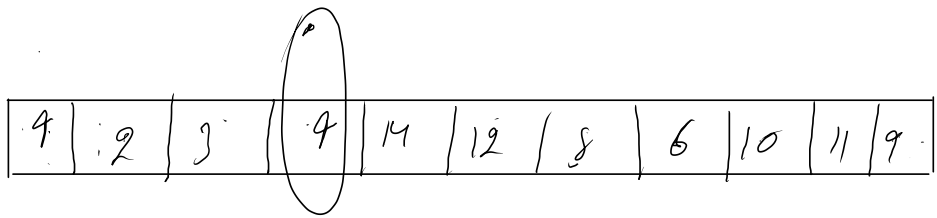
9

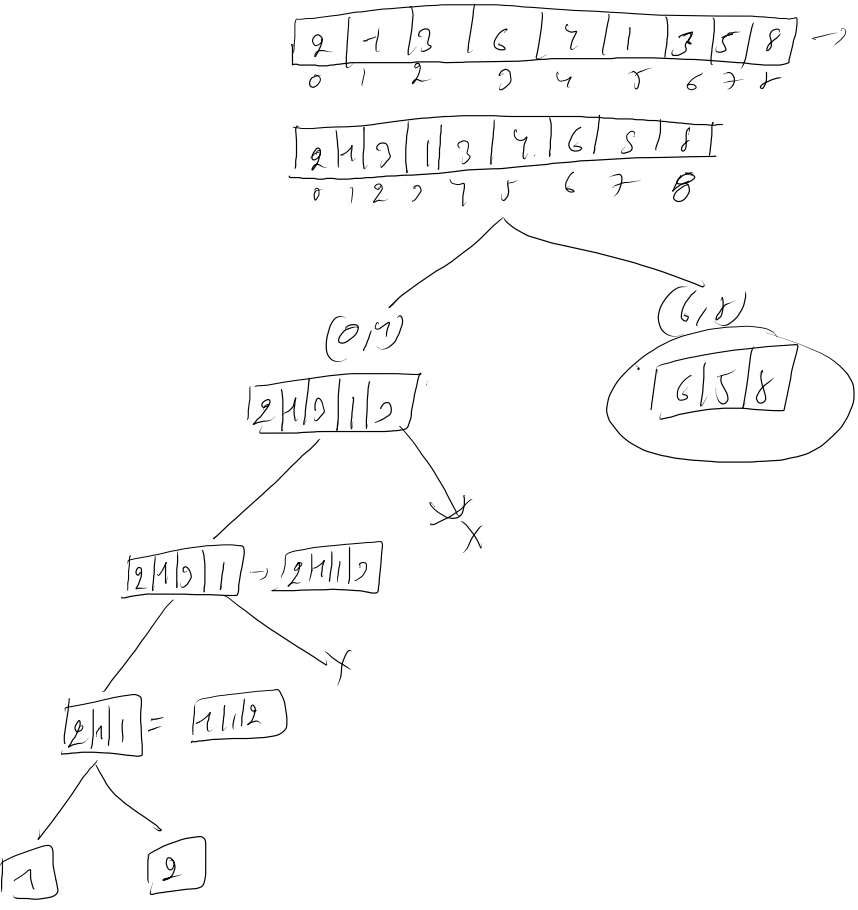
index = 6

data = 9

index = (n-1)

data = arr(x)





worst → (mid is always biggest element in array)

$T(n)$

```python
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)
```

Avg

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

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

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

$N\log_2(N)$

$N, N/2, N/2, \dots$

$N\log_2(N) \rightarrow N, N/2, N/4, \frac{N}{2^2}, \frac{N}{2^3}, \frac{N}{2^4} \dots$

Worst

$T(N) = N + T(N-1)$

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

$T(N-2) = (N-2) + T(N-3)$

$\vdots$

$T(1) = 1 + T(0)$

$T(N) = 1 + 2 + 3 + \dots + (N-2) + (N-1) + N$

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

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