

O/P:- 1 5 9 10 11 12 8 4
3 2 6 7

1. take pointers

2. first loop:- [rowmin → rowmax]
[colmin] Colmin++

3. Second loop:-

[colmin → colmax]

[rowmax]

rowmax--

3. Third loop:-

[rowmax → rowmin]

colmax--

4. [rowmin] } rowmin++
[colmax → colmin]

Sorting algo

① Basic

1. Selection sort

2. Bubble sort

3. Insertion sort

② Advance :-

1. Merge sort

2. Quick sort

1 5 9

4 3 6

↓ Insertion sort

- picks all elements one by one from unsorted array region & insert them at their correct position in sorted region

1 5 9 4 3 6

1 < 9

1 < 5

key = 9

9 > 5

1 3 4 5 6 9

key = 4

key < arr[i]

key = 6

4 < 9

6 < 9

6 > 5

6 < 5

4 > 5

key = 3

$n \rightarrow n-1$

$\frac{3 < 9}{3 < 5}$ $\frac{3 < 4}{\boxed{3 \ 7 \ 1}}$

TC \Rightarrow Best $\rightarrow O(n)$
Worst $\rightarrow O(n^2)$

$O(1) \Rightarrow$ Const
 $O(n) \Rightarrow$ Lin
 $O(n^2) \Rightarrow$ Quad
 $O(n \log n) \Rightarrow$ GRA
 $O(\log n) \Rightarrow$ Polynomial

Q Polynomial

$n! \Rightarrow n^n$ } Exponential

2 5 3 8 4 6 Bubble Sort

I 3 5 8 4 6
 3 5 8 4 6
 3 5 4 8 6
 3 5 4 6 8

II 3 5 4 6 8 III 3 4 5 6 8
 3 5 4 6 8
 3 4 5 6 8
3 4 5 6 8

is swapped
 false
break

$n-1$

TC $\rightarrow O(n)$
 $O(n^2)$

3 Selection Sort

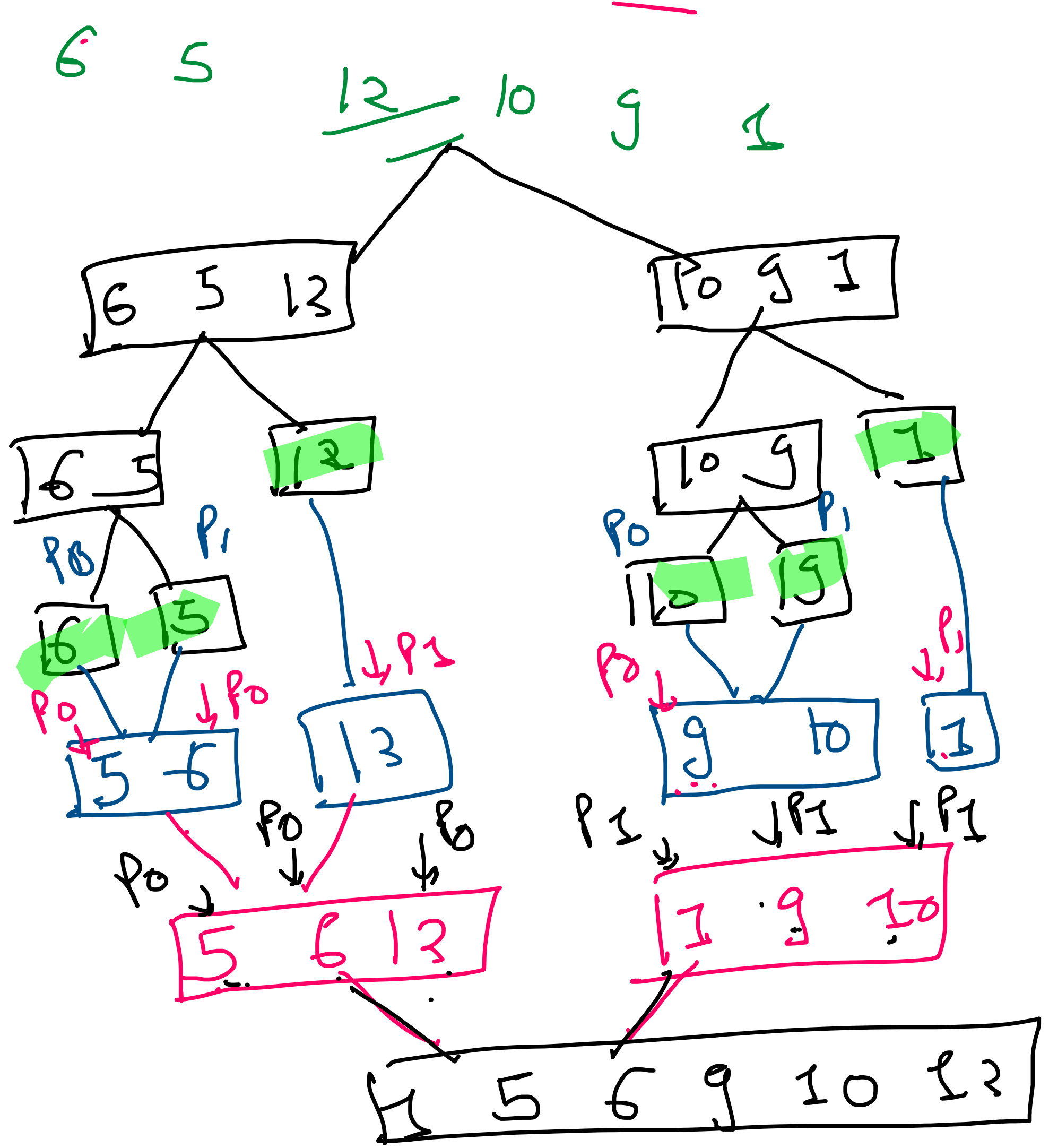
20 12 10 9 15 3

Select \Rightarrow elem
 \hookrightarrow get start of unsorted region

2 ; 12 10 9 15 20
 2 9 ; 10 12 15 20
 2 9 10 ; 12 15 20
 2 9 10 12 ; 15 20

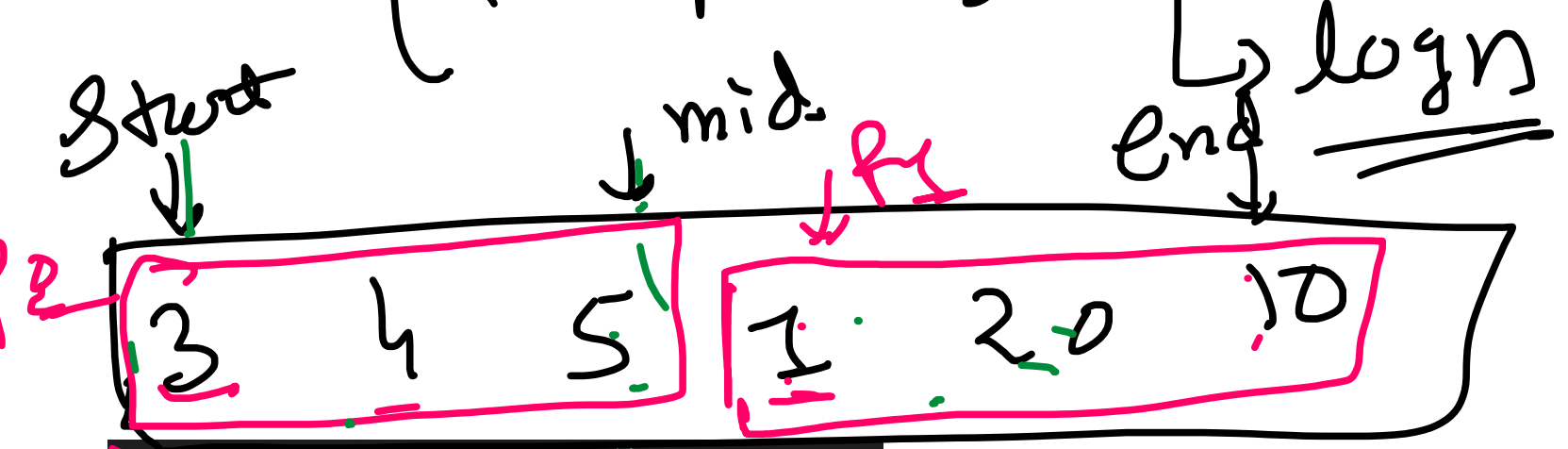
2 9 10 12 15 20
 2 9 10 12 15 20
 2 9 10 12 15 20

Merge Sort: } Divide & Conquer



Space comp

- 1. Extra array space } $O(N)$
- 2. Recursion call. } $\log n$



```

function merge(arr, start, mid, end) {
  let m1 = mid - start + 1;
  let m2 = end - mid;

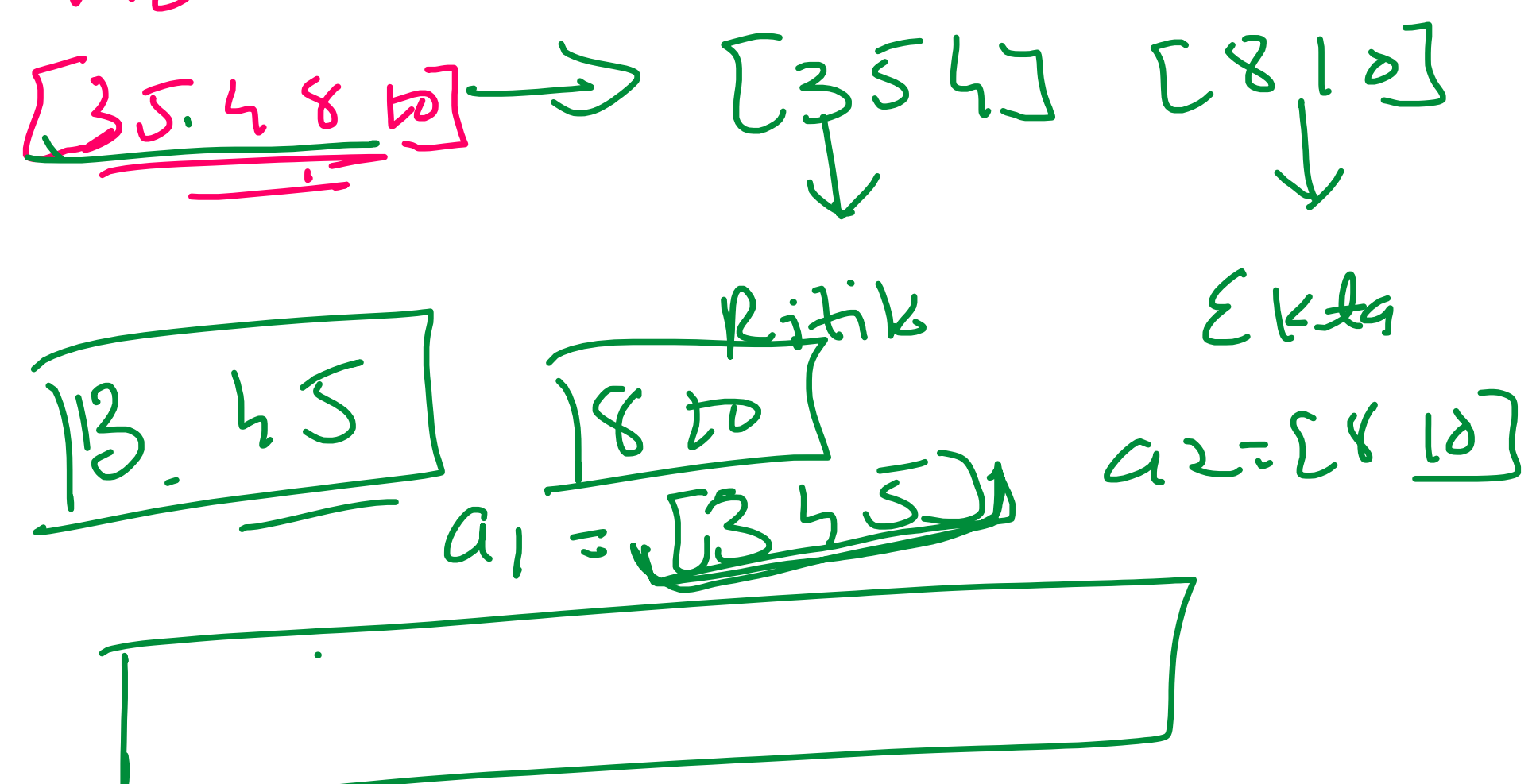
  let a1 = new Array(m1);
  let a2 = new Array(m2);

  for (let i = 0; i < m1; i++) {
    a1[i] = arr[start + i];
  }
  for (let j = 0; j < m2; j++) {
    a2[j] = arr[mid + 1 + j];
  }

  let p1 = 0; // a1
  let p2 = 0; // a2
  let k = start; // res
  
```

$mid - start + 1$
 $2 - 0 + 1 \Rightarrow 3$
 $end - mid$
 $5 - 2 \Rightarrow 3$

Visual



Q2 Quick Sort: } Partitions

5 6 9 1 4 3 2 7

