

Insertion sort

Insertsort(A)

{ for (j = 2 to A.length)

{ key = A[j]

i = j - 1;

while (i > 0 && A[i] > key)

{

A[i+1] = A[i];

i = i - 1;

}

A[i+1] = key;

}

Best time case -

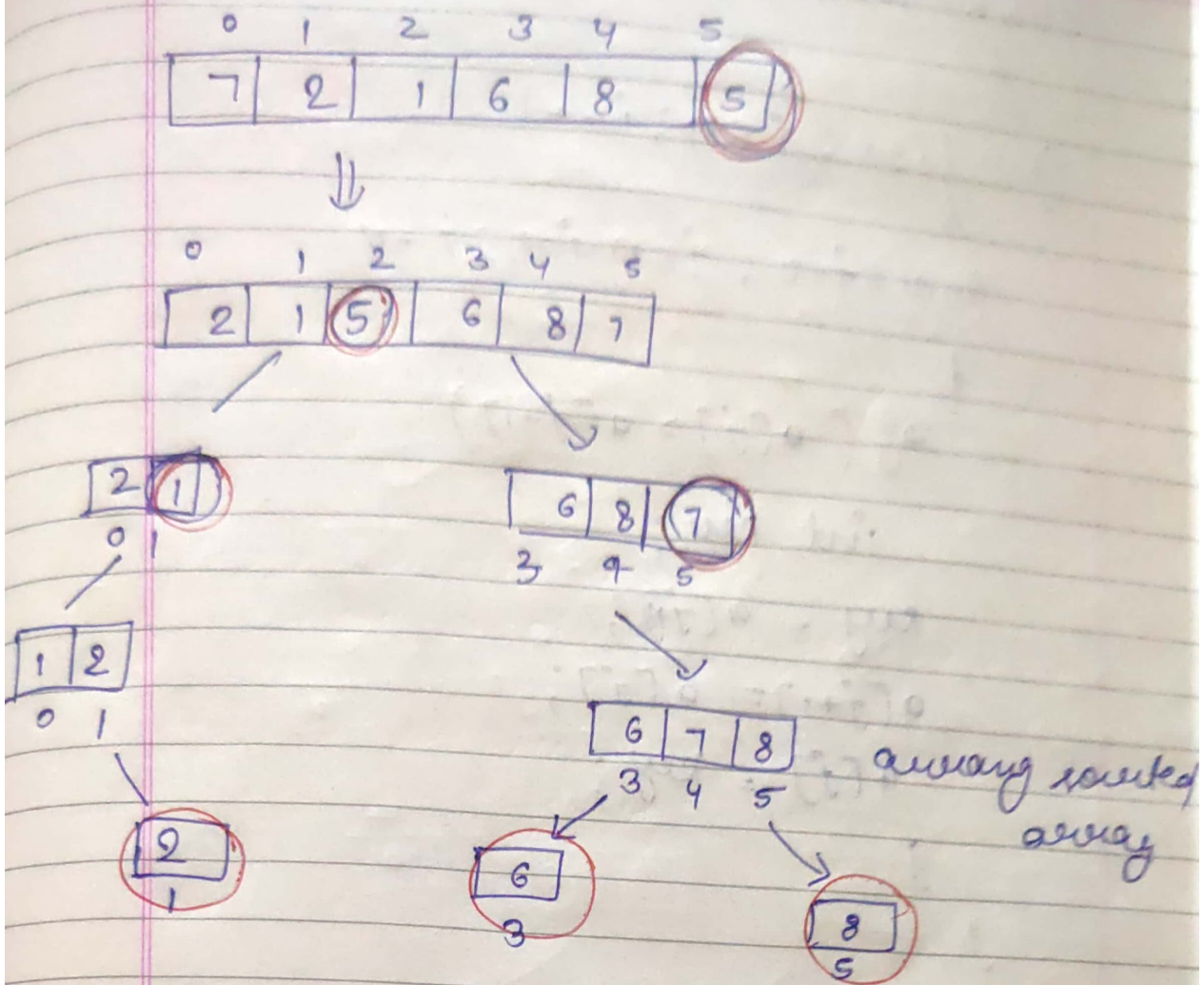
①

i = 4

	<div>4 2 1 3 6</div>					
i =	0	1	2	3	4	5
elem.		1 2 4	2 2 4	3 1 4	<u>3</u> 4	6 6
j			2	3	4	5
key			2	1	3	6
comp			1	1+1	1	1
move			1	1+1	1	0
			<u>2</u> (1)	<u>2</u> (2)	<u>2</u> (1)	<u>1</u>

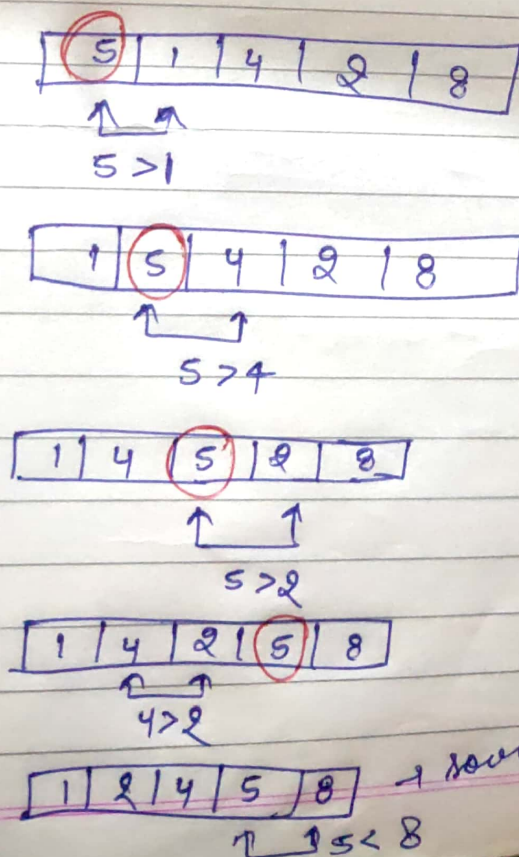
$$2(1+2+1+0+...+n-1) = 2(n-1)/2 = O(n^2)$$

Quick sort



Bubble sort →

swapping



Bubblesort (A)

{
for ($i=0$; $i < n$; $i++$)

{
for ($j=0$; $j < n-i-1$; $j++$)

{
if ($a[j] > a[j+1]$)

int key;

key = $a[j+1]$;

$a[j+1] = a[j]$;

$a[j] = \text{key}$;

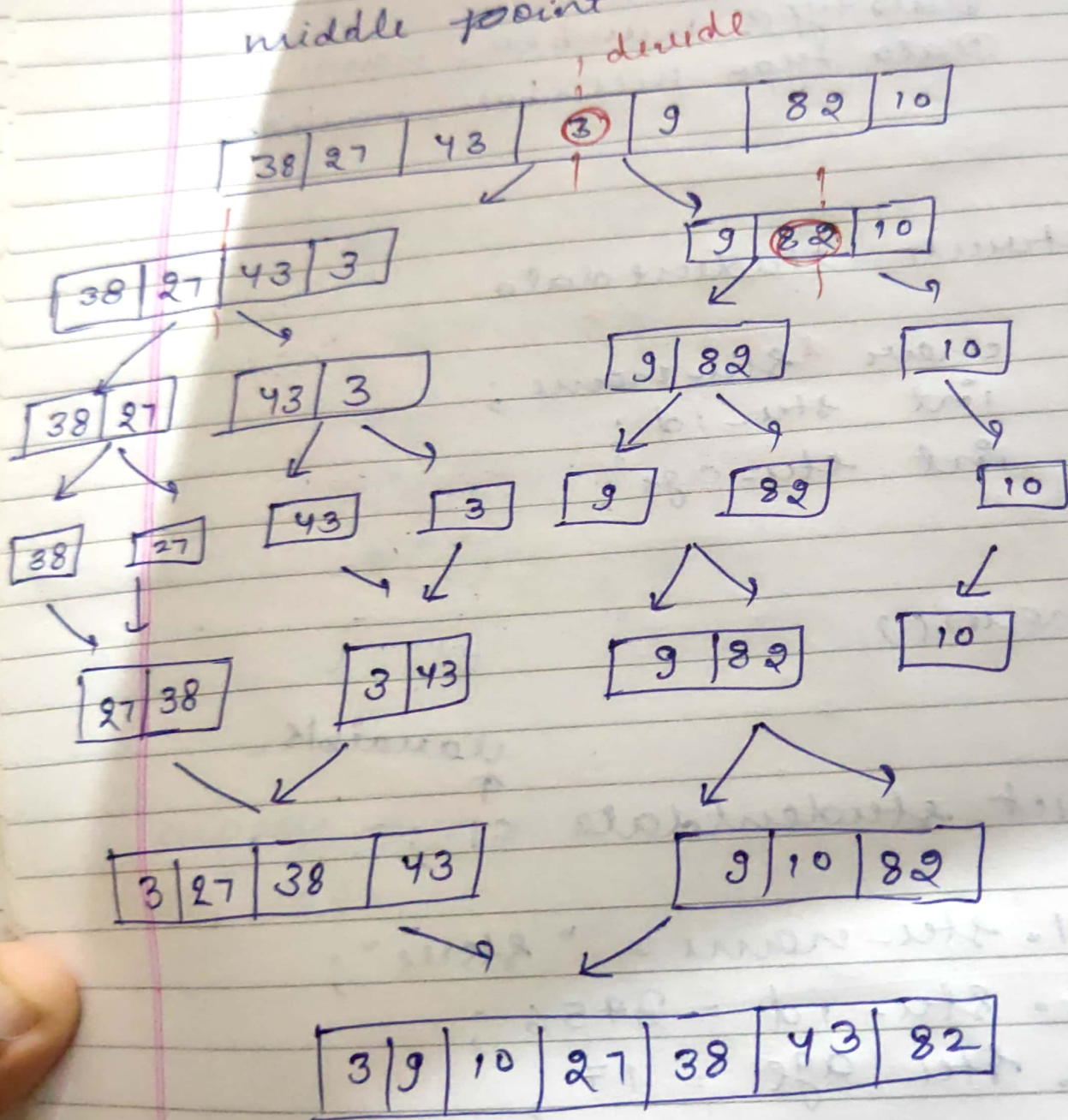
}

}

suppose $\rightarrow n = 5$

Merge sort -

↓
it divide array in two sub arrays.
middle point.



time complexity $\rightarrow O(N \log N)$