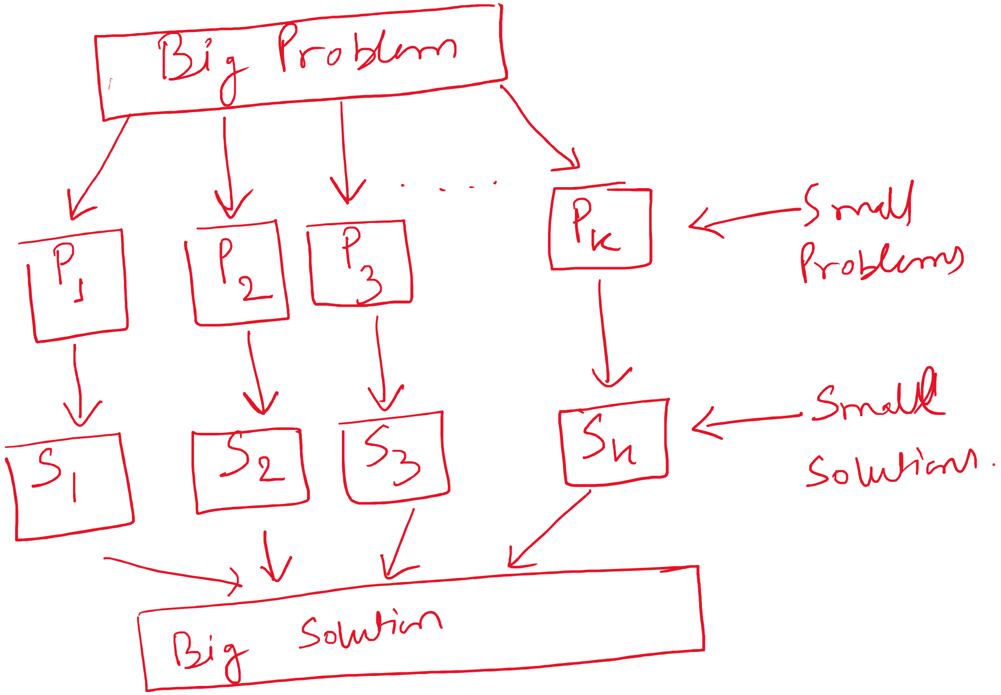
Divide and conquer Technique – Technique to design an algorithm



Eg –

$$\underbrace{1+2+3+4}_{=10}$$

$P: 1+2+3+4$

$P_1 = 1+2 \rightarrow S_1 = 3$

$P_2 = 3+4 \rightarrow S_2 = 7$

$S_1 \quad S_2$

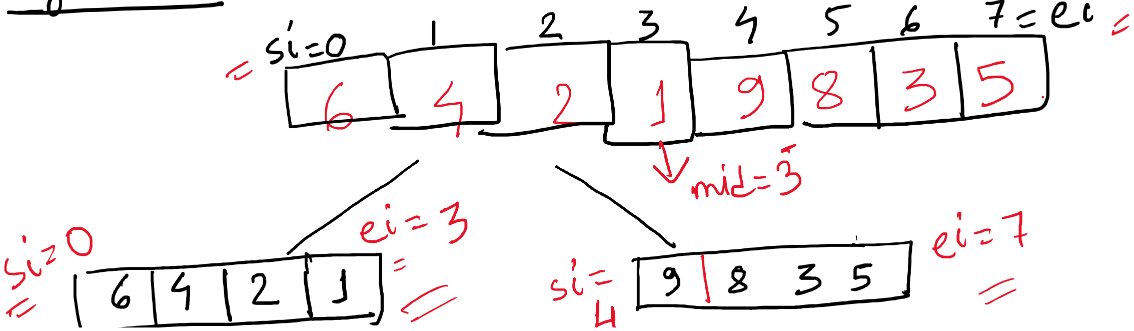
$3+7=10$

$$mid = \left(\frac{si+ei}{2}\right)$$

$$= \left(\frac{0+7}{2}\right) = 3.5$$

$\boxed{mid = 3}$

Merge Sort – Based on Divide & Conquer Technique.



si=0	1	2	3	4	5	6	7=ei
6	4	2	1	9	8	3	5

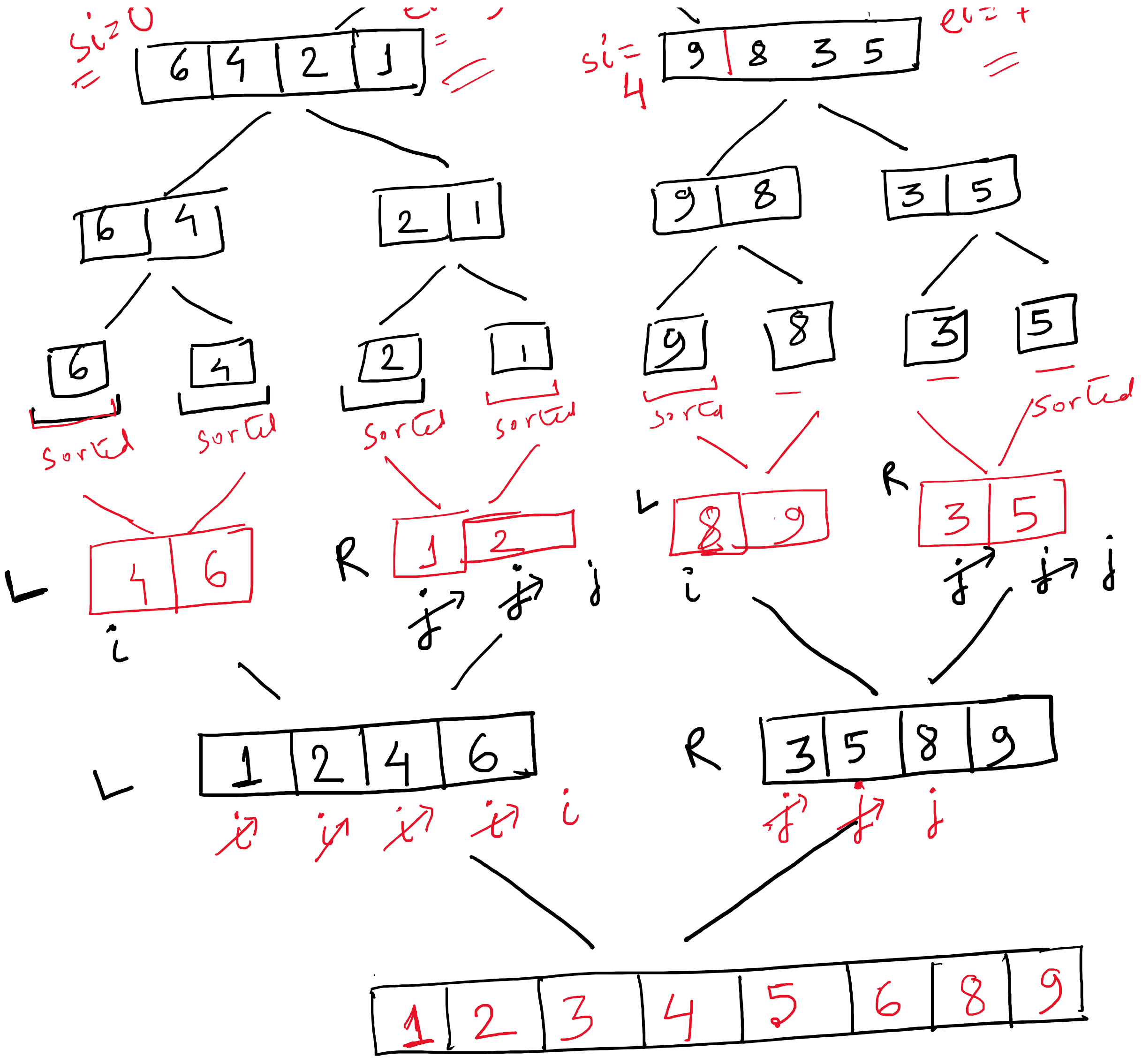
6	4	2	1

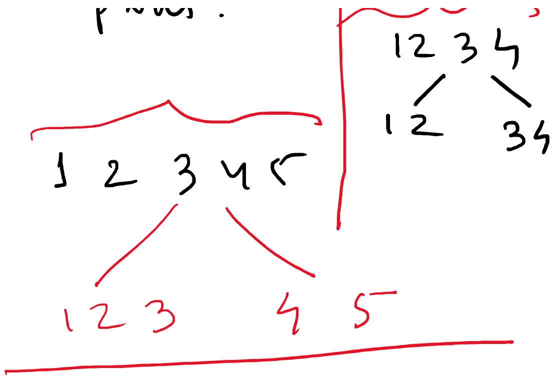
9	8	3	5

Note –

① Repeatedly divide the array into 2 equal parts.

1234





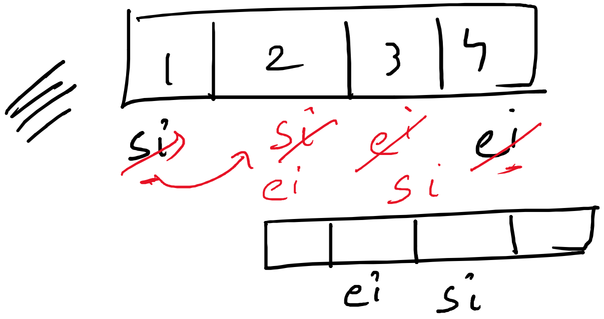
② Combine 2 sorted arrays into 1 sorted array.

$L[i] > R[j]$

$L(si, m)$ $R(m+1, ei)$

Code:

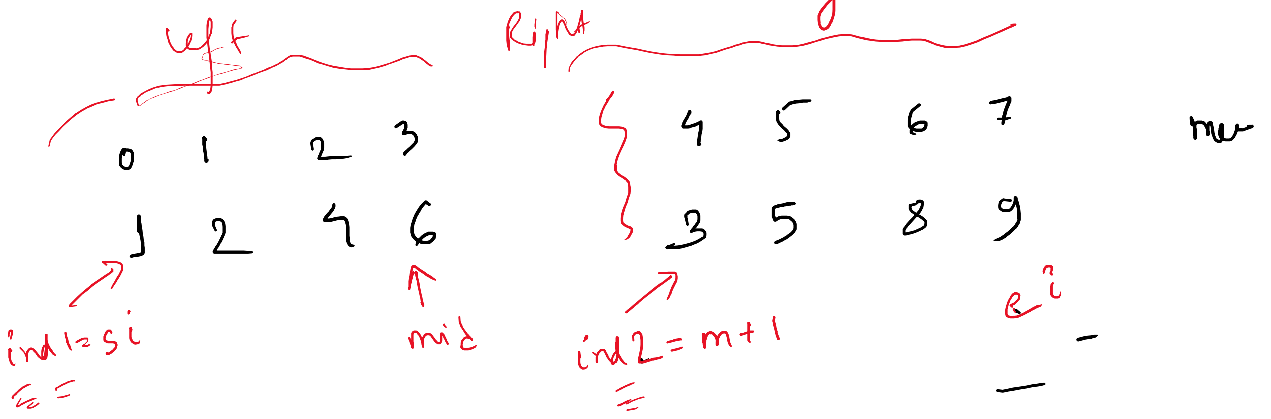
```
main() {
        divide(arr, si=0, ei=n-1);
}

divide(arr, si, ei){
  if(si<ei) {
      mind = (si+ei)/2;
      divide(arr, si, m);   // left array.
      divide(arr, m+1, ei);   // Right array
      merge(arr, si, m, ei);   // left array → si, m
                                  Right array → m+1, ei),
  }
}
```



```
merge (arr, si, m, ei) {   // Left sorted Array
                           // Right sorted Array
                           // Merge into 1 sorted array.
```

// Right sorted array

// Merge into 1 sorted array.



```
int merged[] = new int[ei - si + 1];
```