Dry-Run

Merge sort:

```
int main () {
int arr[] = {64, 34, 25, 12, 22, 11}
```

64	34	25	12	22	11

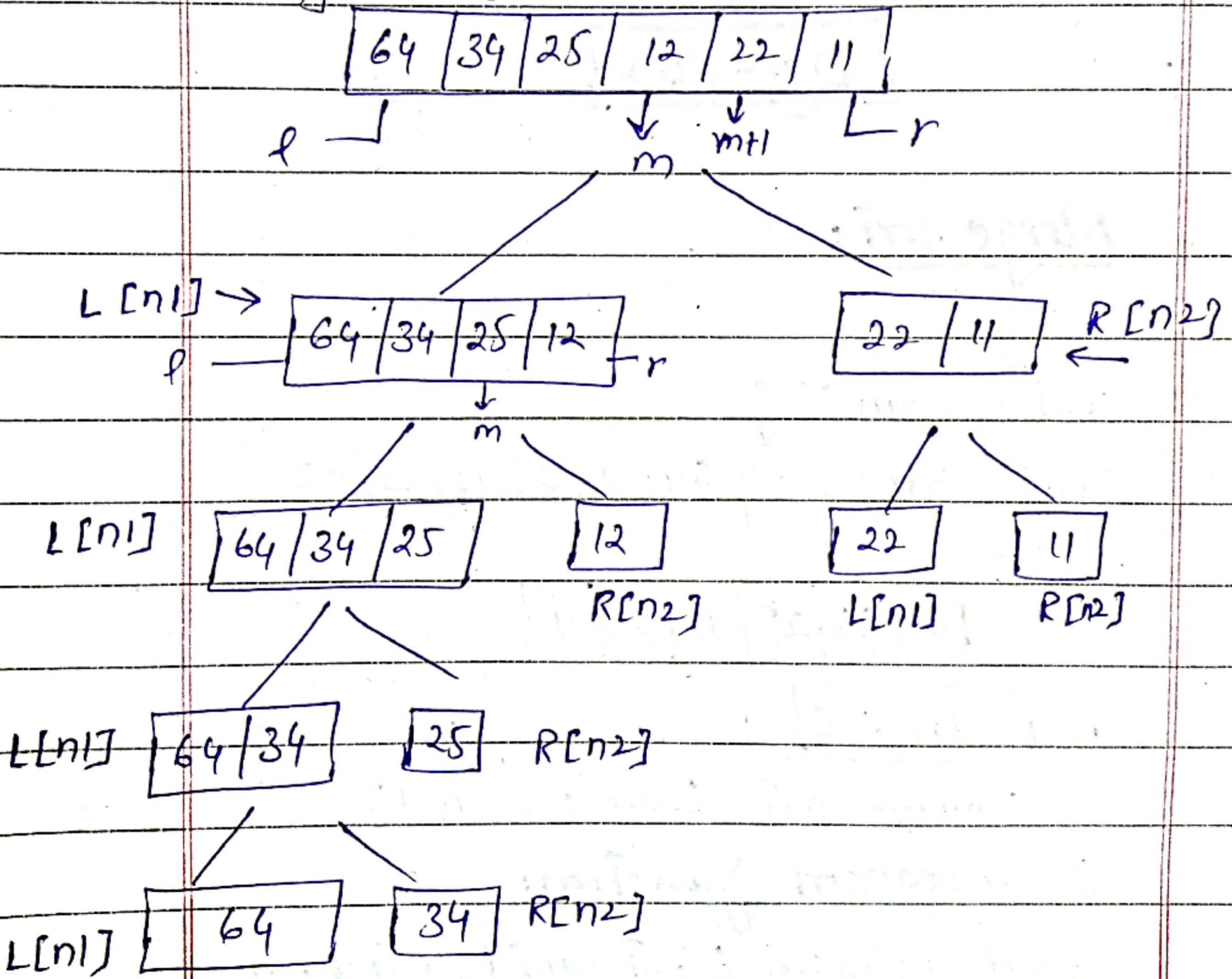
```
int n = 6
    merge sort (arr, 0, n-1)
// mergesort function:
void mergsort (int arr[], int l, int r)
```

64	34	25	12	22	11
l					r

DATE: DAY:

```
if (l < r) — True    0 < 5
{
 int m = l + (r-l)/2
       = 1 + (5)/2
 int m = 3

 merge sort (arr, l, m);
 mergesort( arr, m+1, r);
```



DATE: DAY:

merge (arr , l, m, r) ;

64	34	25	12	22	11

$n1 = 3-0+1 = \underline{4}$

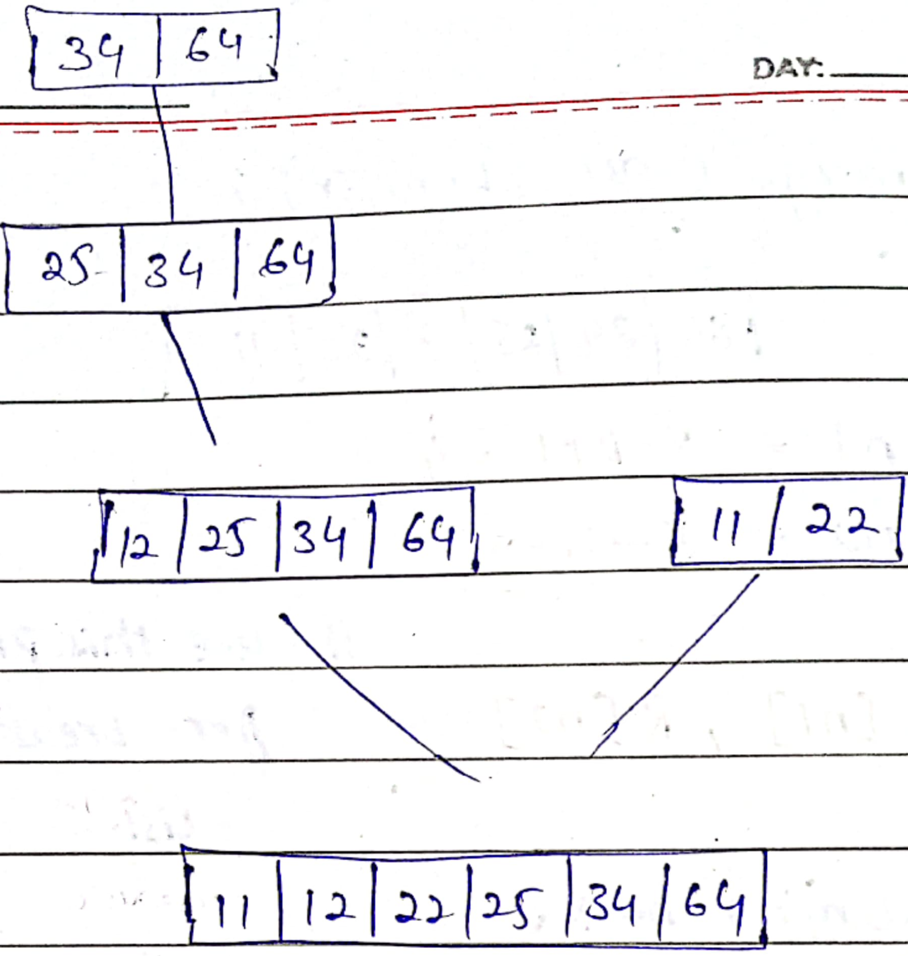
$n2 = 5-3 = 2$

L [n1] , R [n2]   // use this process for breaking whole list.

// Merging back.

```
int  i=0 , j=0 , K=l.
              0<1           0<1
   while ( i<n1 && j<n2) {
      if ( L[i] <= R[j]){          64 <= 34
           arr[K] = L[i]
           i++;
      }
   else {
         arr[K] = R[j]          arr[0] = 34
           j++;                     j++
      }
     K++                          K++.
}
```

34	64



Sorted array: