MERGE SORT

ALGORITHM IN PYTHON

TABLE OF CONTENTS

MERGE SORT

Merge Sort is one of the most popular sorting algorithms that is based on the principle of **Divide and Conquer Algorithm.**

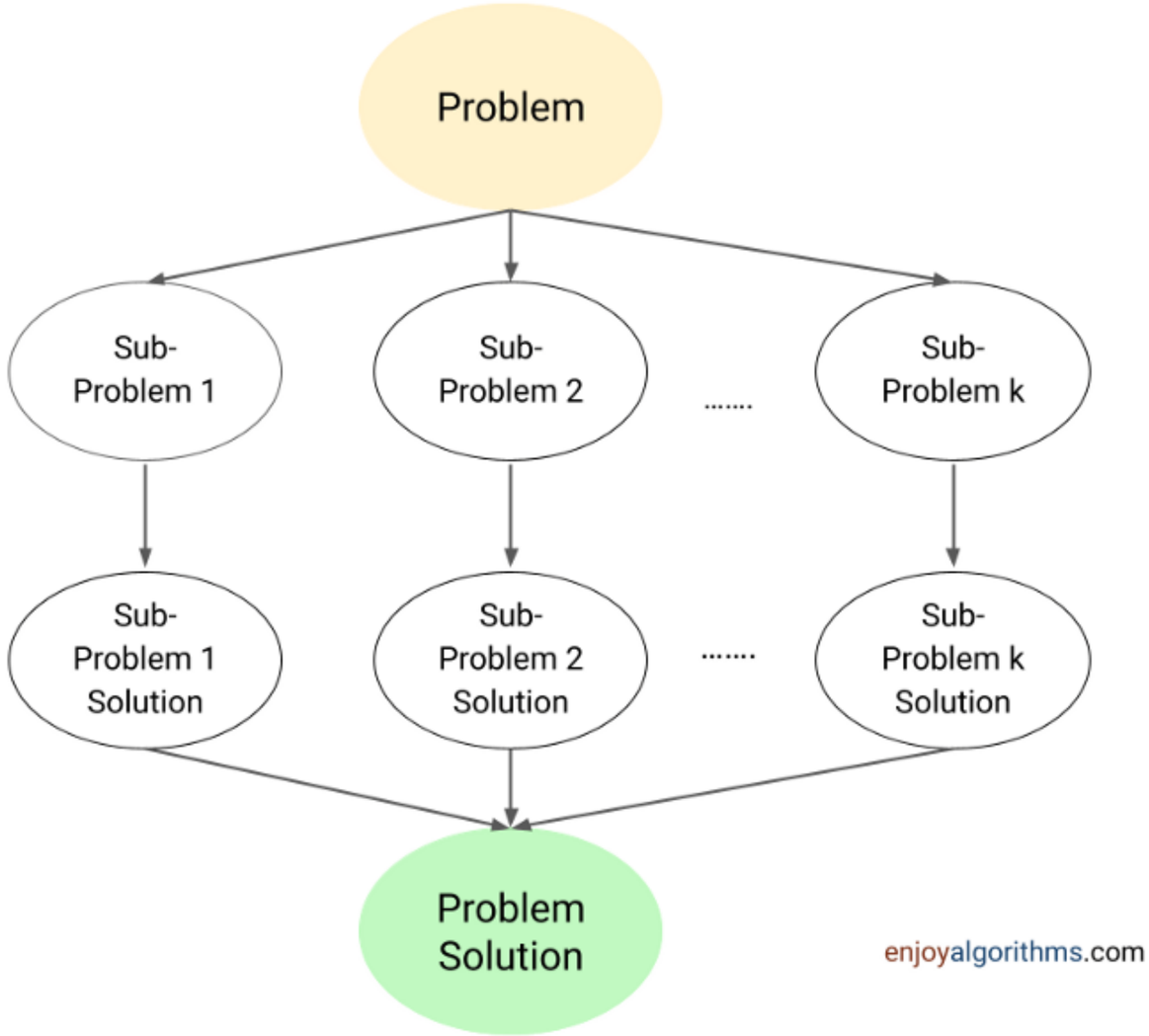
Here, a problem is divided into **multiple sub-problems**. Each sub-problem is solved individually. Finally, sub-problems are combined to form the final solution.

DIVIDE AND CONQUER

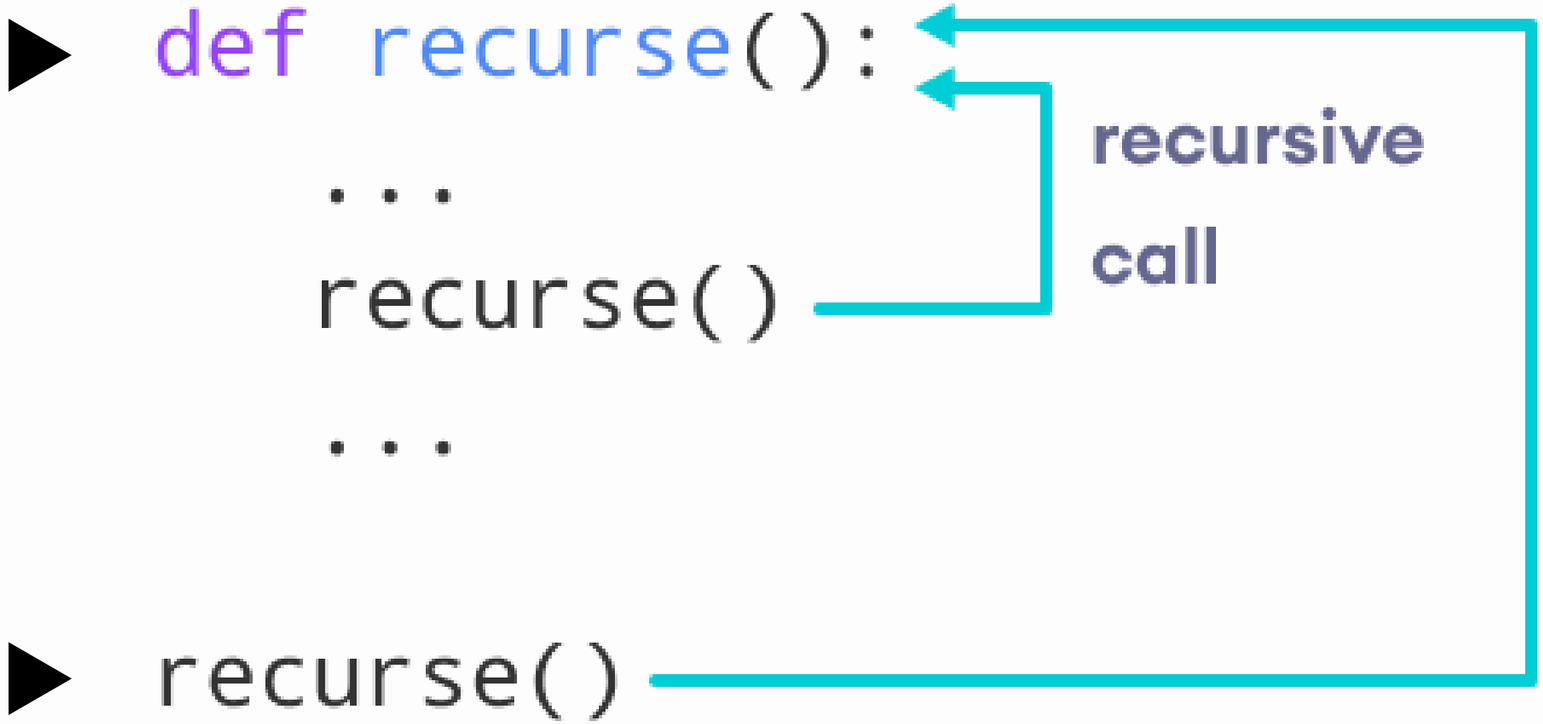
Divide
Dividing the problem into smaller sub-problems

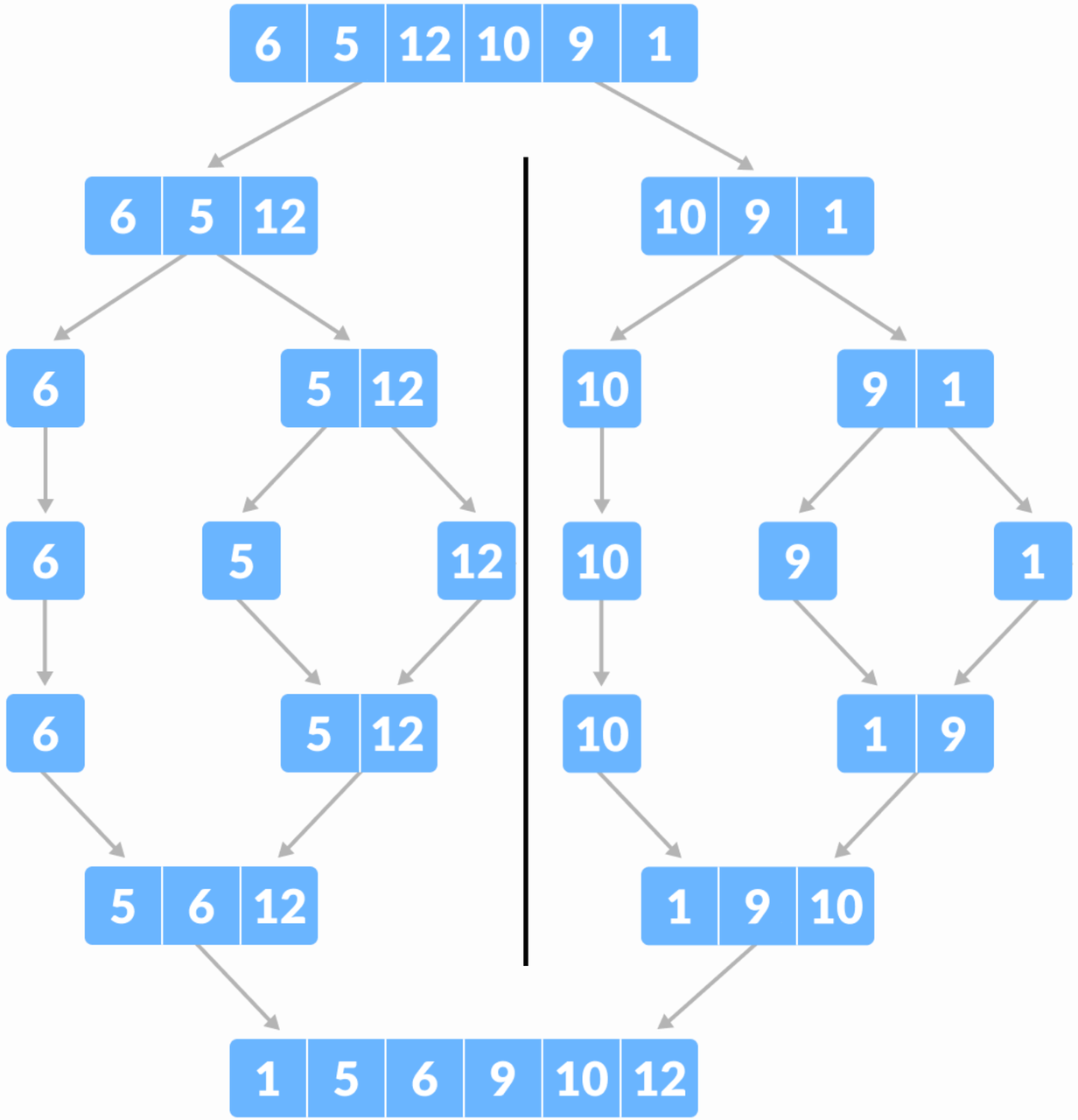
Conquer
Solving each sub-problems recursively

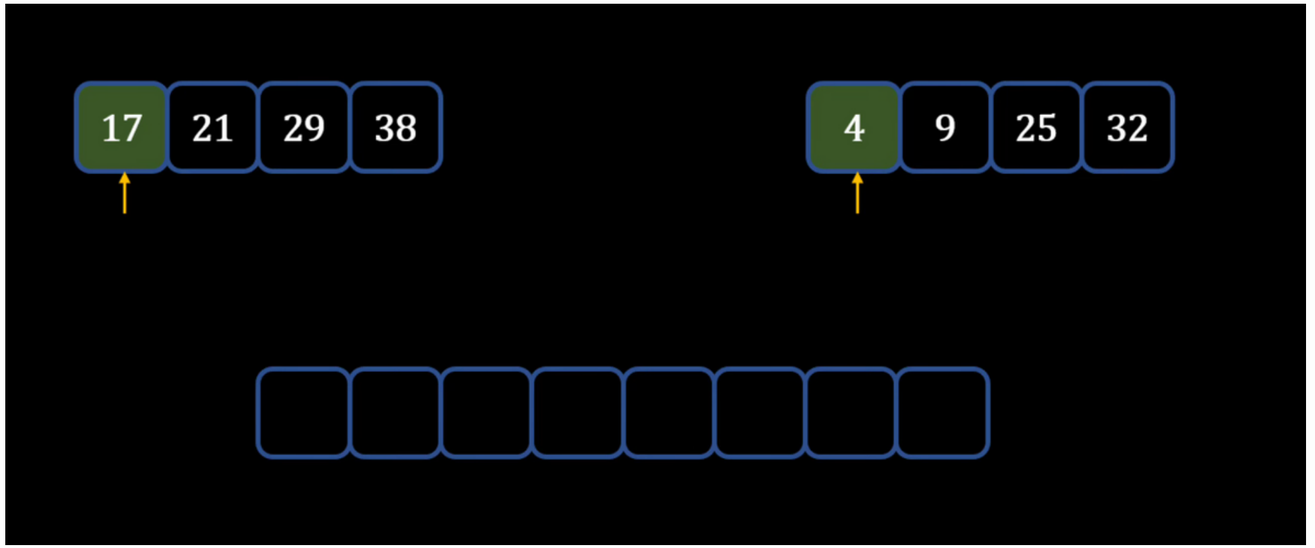
Combine
Combining sub-problem solutions to build the original problem solution

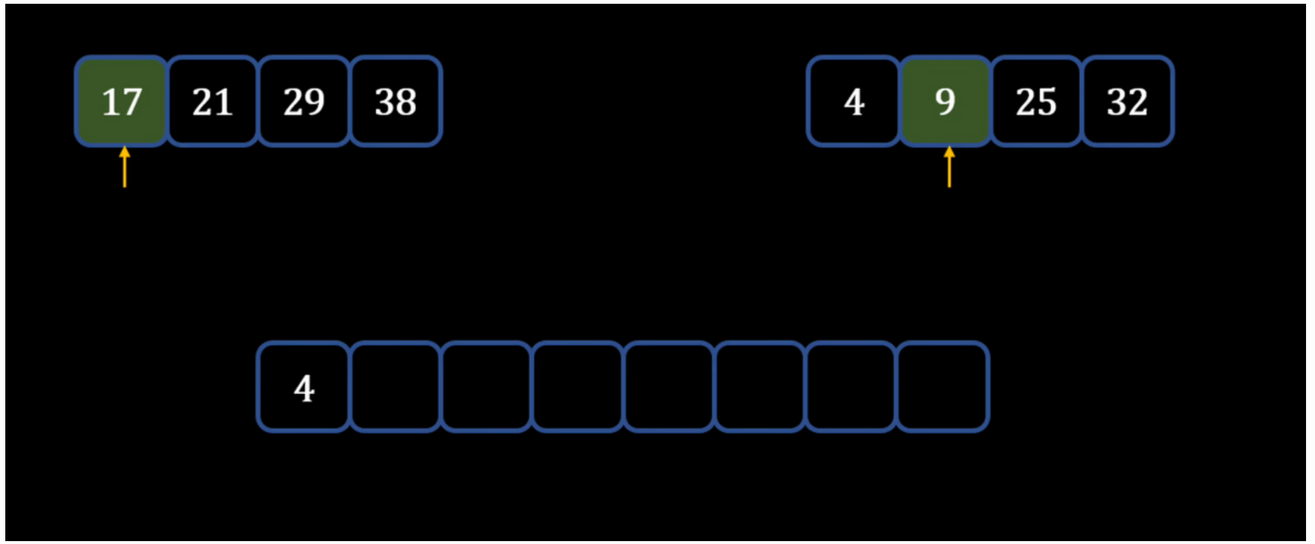


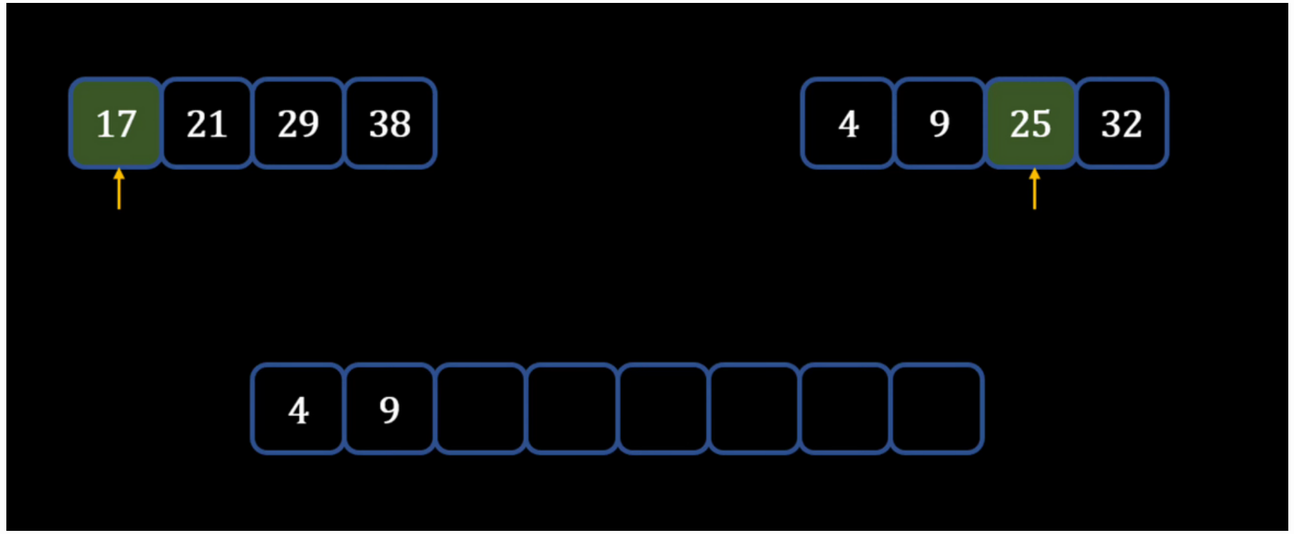
RECURSION

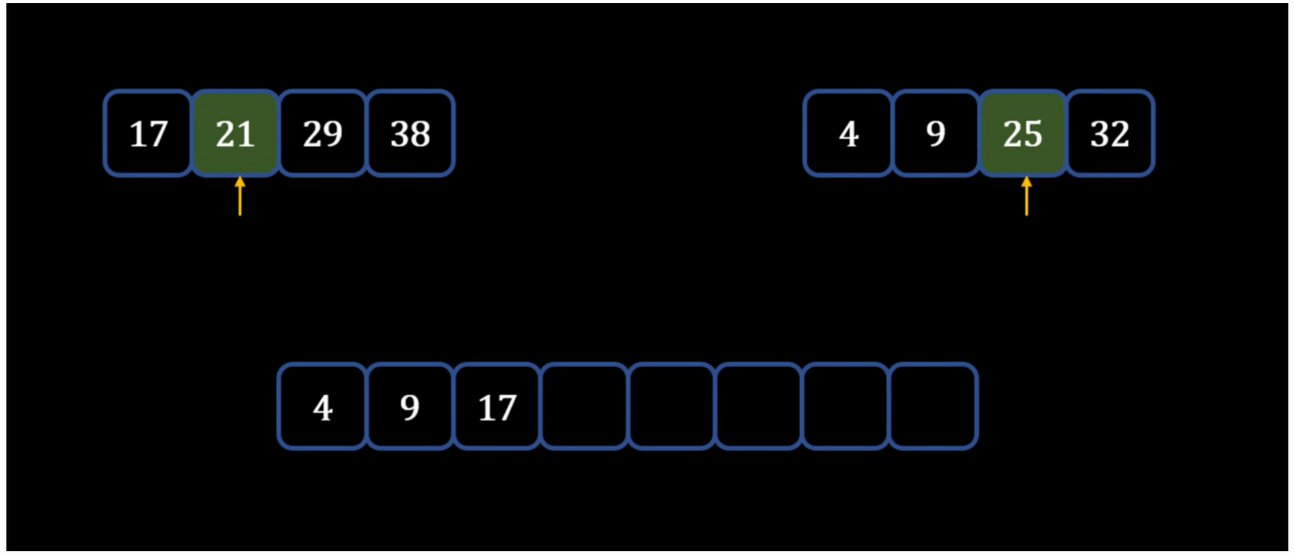

def recurse():
...
recurse()
...
recurse()
recursive
call


6 5 12 10 9 1
6 5 12
10 9 1
6
5 12
10
9 1
6
5
12
10
9
1
6
5 12
10
1 9
5 6 12
1 9 10
1 5 6 9 10 12

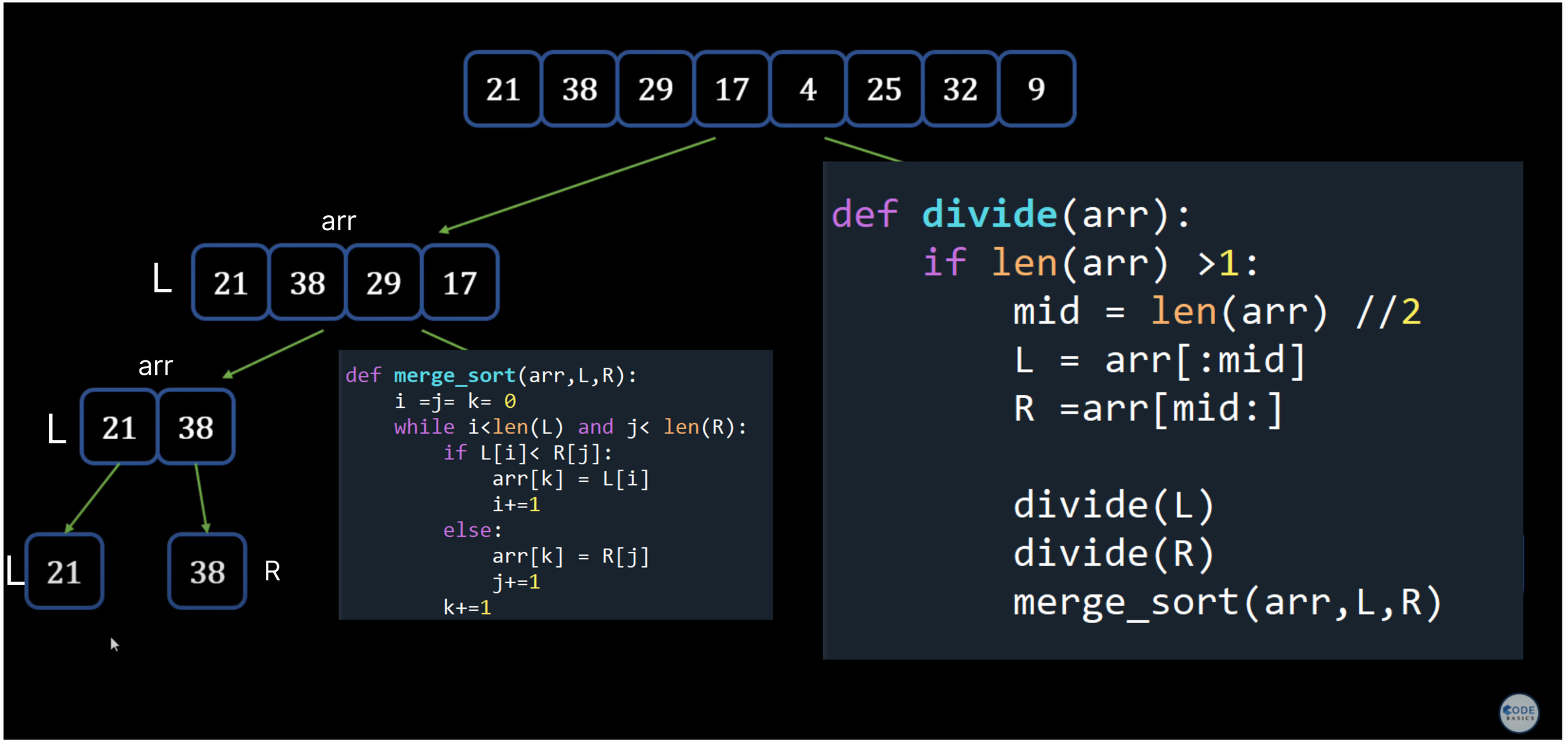

17
21
29
38
4
9
25
32

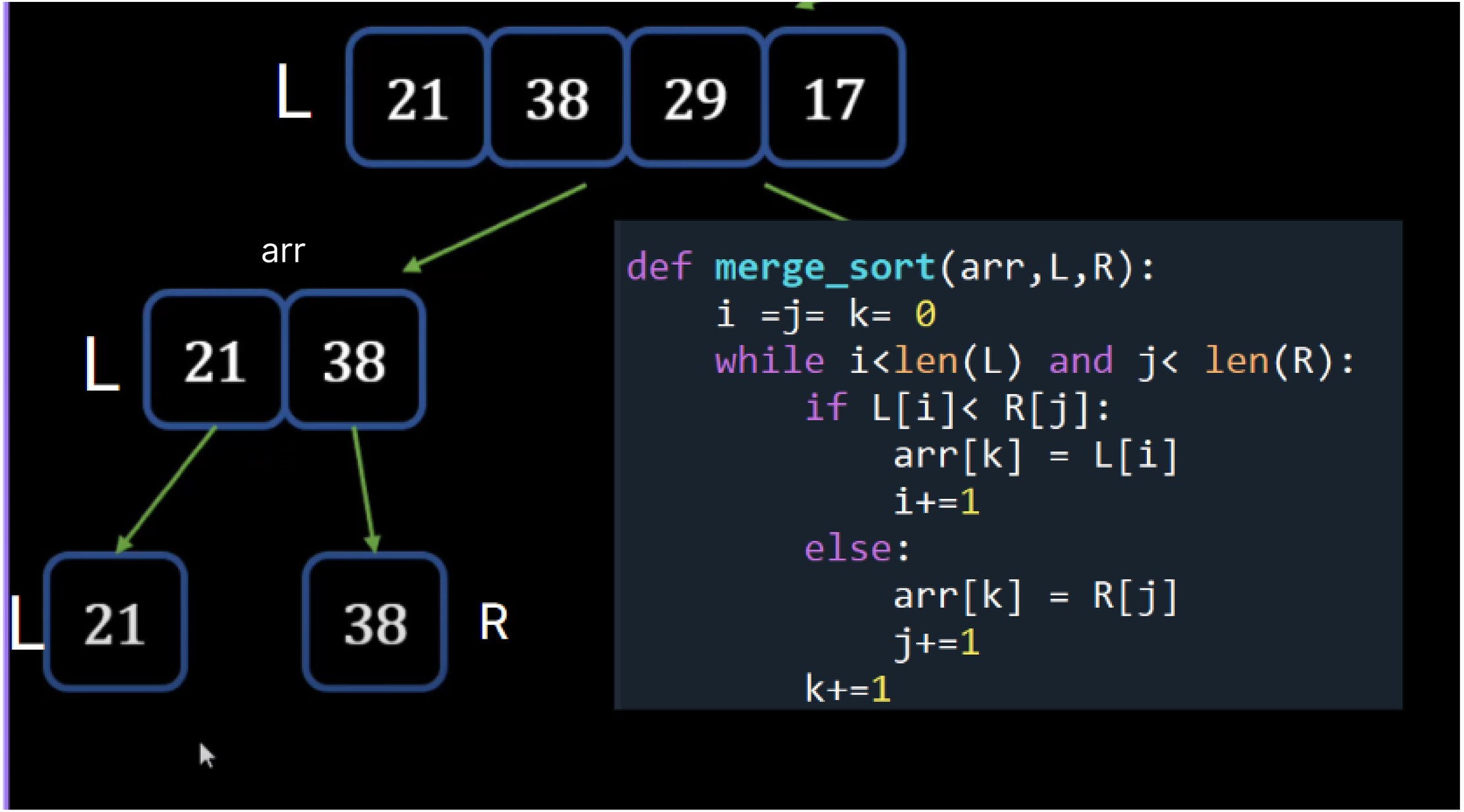

17
21
29
38
4
9
25
32
4


17
21
29
38
4
9
25
32
4
9


17
21
29
38
4
9
25
32
4
9
17


17 21 29 38
4 9 25 32
4 9 17 21 25 29 32 38


21 38 29 17 4 25 32 9
arr
L
21 38 29 17
arr
L
21 38
L
21
38
R
def merge_sort(arr,L,R):
i =j= k= 0
while i<len(L) and j< len(R):
if L[i]< R[j]:
arr[k] = L[i]
i+=1
else:
arr[k] = R[j]
j+=1
k+=1
def divide(arr):
if len(arr) >1:
mid = len(arr) //2
L = arr[:mid]
R =arr[mid:]
divide(L)
divide(R)
merge_sort(arr,L,R)




```python
def merge_sort(arr,L,R):
    i =j= k= 0
    while i<len(L) and j< len(R):
        if L[i]< R[j]:
            arr[k] = L[i]
            i+=1
        else:
            arr[k] = R[j]
            j+=1
        k+=1
```