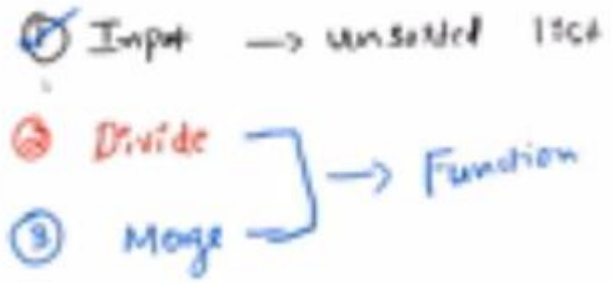
MergeSort

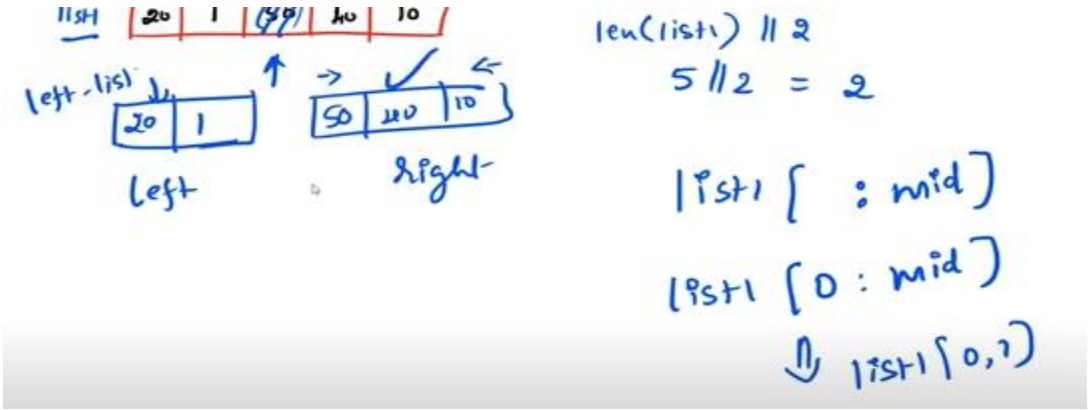
1. split the unsorted list.
2. compare each of the elements and group them
3. repeat step 2 untill whole list is merged and sorted.

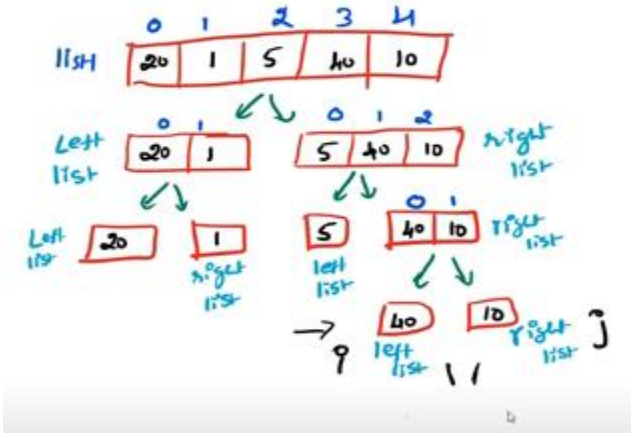



```
num = int(input("how many elements you want in list:"))
list1 = [int(input()) for x in range(num)]

def mergesort(list1):
    if len(list1)>1:
        mid = len(list1)//2
        left_list = list1[:mid]
        right_list = list1[mid:]
        mergesort(left_list)
        mergesort(right_list)
```

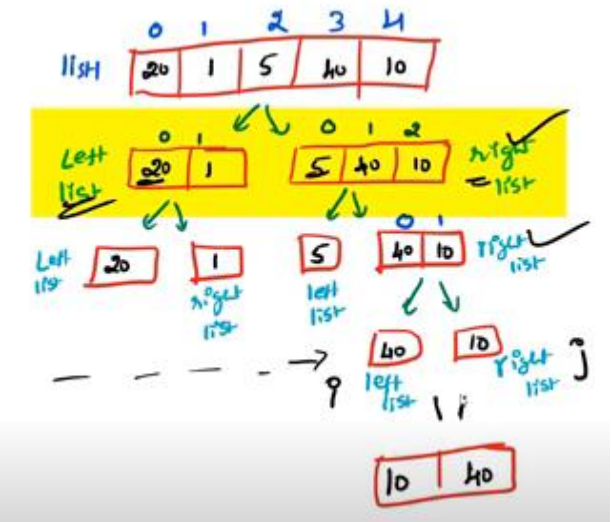
perintah di atas adalah rekursif, yaitu fungsi utk membagi list hingga single element.





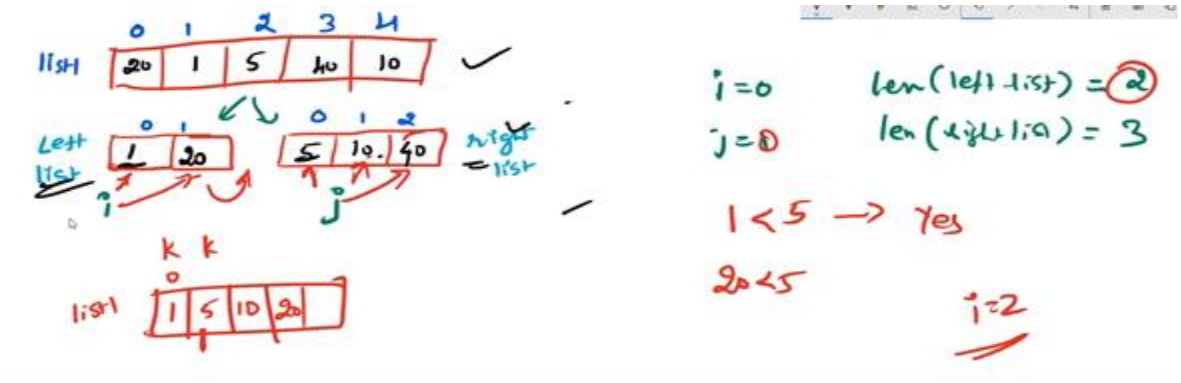
```
i=0
j=0
k=0
if left_list[i]<right_list[j]:
    list1[k] = left_list[i]
else:
    list1[k] = right_list[j]
```

krn dilakukan scr berulang pada stage k=0 hingga tertentu maka mgunakn perulangan ‘while’



```
i=0
j=0
k=0
while i<len(left_list) and j<len(right_list):
    if left_list[i]<right_list[j]:
        list1[k] = left_list[i]
        i=i+1
        k=k+1
    else:
        list1[k] = right_list[j]
        j=j+1
        k=k+1
```

Code utk merging



Utk meletakkn nilai/elemen list1 maka kita butuh mlakukn increment ‘k’.

```
while len(left_list)>i:
    list1[k] = left_list[i]
    i=i+1
    k=k+1
while len(right_list)>j:
    list1[k] = right_list[j]
    j=j+1
    k=k+1
```

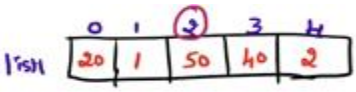
code utk memeriksa apakah ada nilai yang tertinggal (tersisa) atau tidak. di sublist kiri, code ini untuk memeriksa apakah ada nilai yang tersisa di sublist kanan. jika ada nilai yang tersisa maka kami akan menambahkannya ke list1 (begitu pula sebaliknya utk ‘while’ brktnya)

```
num = int(input("how many elements you want in list:"))
list1 = [int(input()) for x in range(num)]
mergesort(list1)
print("sorted list is:",list1)
```

```
how many elements you want in list:5
20
1
50
40
10
sorted list is: [1, 10, 20, 40, 50]
```

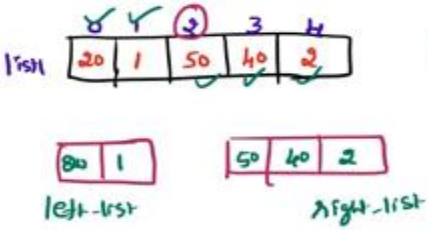
```
how many elements you want in list:6
1
75
42
15
33
1
sorted list is: [75, 42, 33, 15, 1, 1]
```

Penjelasan code utk 'divide' hingga diperoleh single element



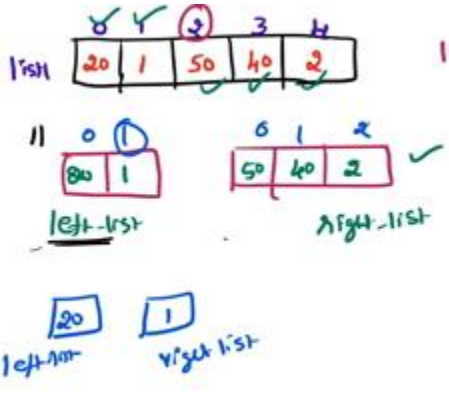
len(list1) =5

```
def mergesort(list1):
    if len(list1)>1:          #basecase
(2)   mid = len(list1)//2
→   left_list = list1[:mid]
    right_list = list1[mid:]
    mergesort(left_list)
    mergesort(right_list)
```



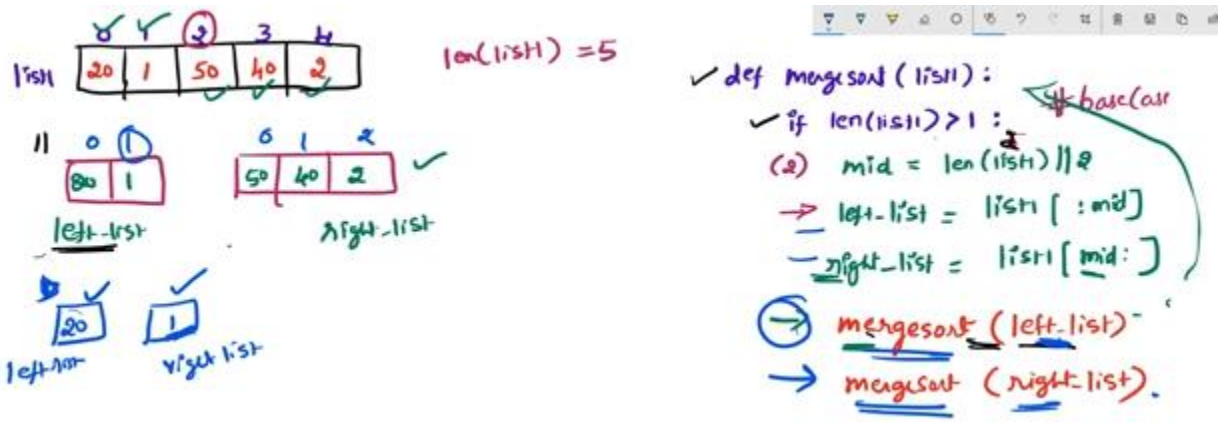
len(list1) =5

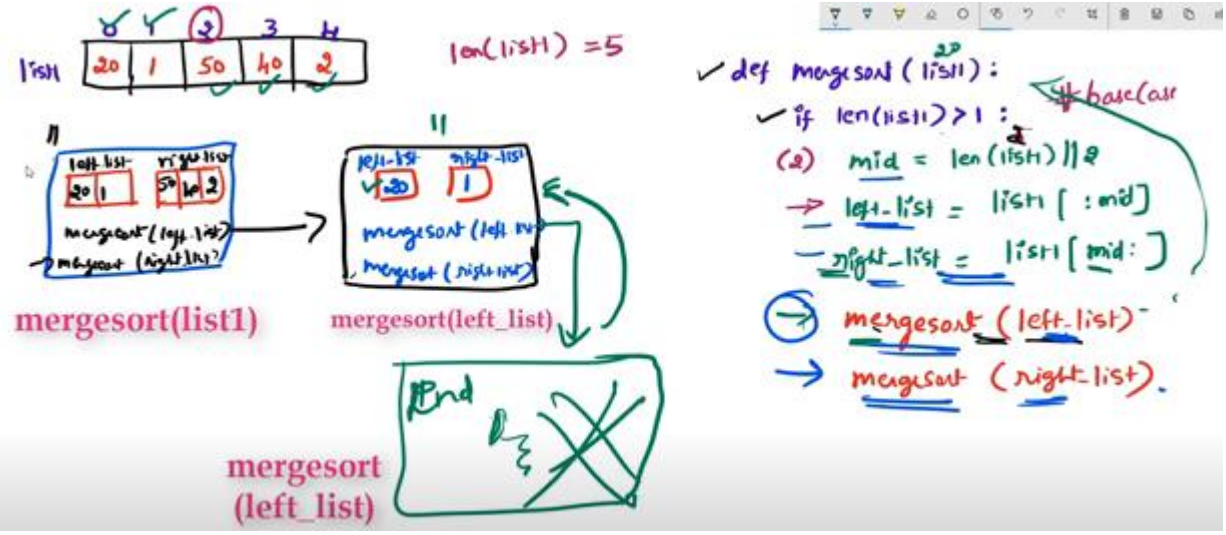
```
def mergesort(list1):
    if len(list1)>1:          #basecase
(2)   mid = len(list1)//2
→   left_list = list1[:mid]
    right_list = list1[mid:]
→   mergesort(left_list)
    mergesort(right_list)
```

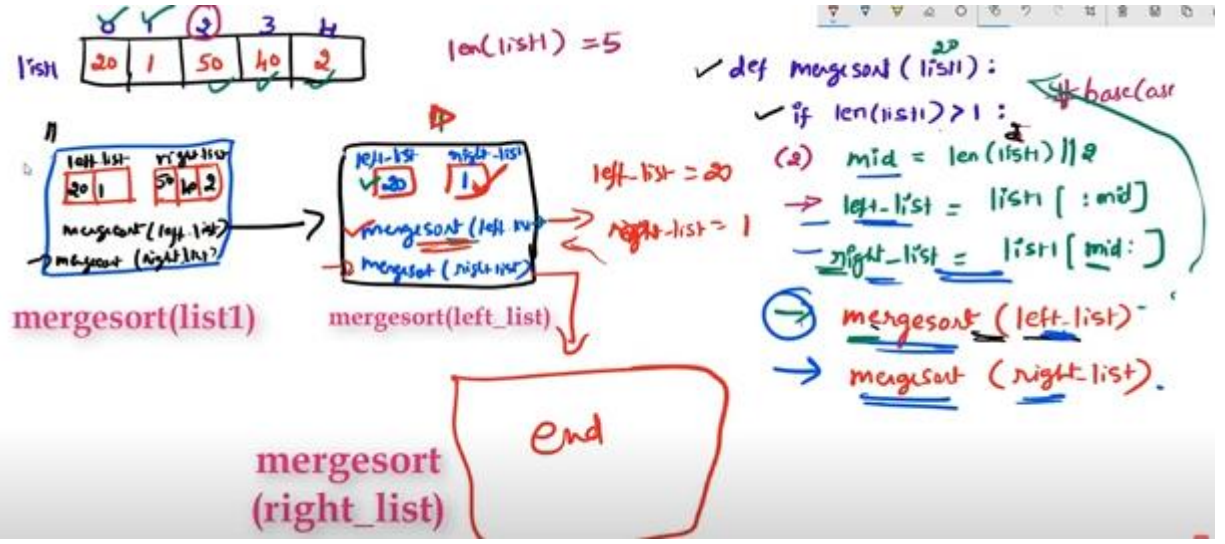


len(list1) =5

```
✓def mergesort(list1):
✓   if len(list1)>1:          #basecase
(2)   mid = len(list1)//2
→   left_list = list1[:mid]
    right_list = list1[mid:]
→   mergesort(left_list)
    mergesort(right_list)
```


list
20 1 50 40 2
len(list1) = 5
left_list
right_list
def mergesort (list1):
if len(list1) > 1 :
mid = len(list1) // 2
left_list = list1 [: mid]
right_list = list1 [mid :]
mergesort (left_list)
mergesort (right_list).
base case


list
20 1 50 40 2
len(list1) = 5
mergesort(list1)
mergesort(left_list)
mergesort
(left_list)
End
def mergesort (list1):
if len(list1) > 1 :
mid = len(list1) // 2
left_list = list1 [: mid]
right_list = list1 [mid :]
mergesort (left_list)
mergesort (right_list).
base case


list
20 1 50 40 2
len(list1) = 5
left_list = 20
right_list = 1
mergesort(list1)
mergesort(left_list)
mergesort
(right_list)
End
def mergesort (list1):
if len(list1) > 1 :
mid = len(list1) // 2
left_list = list1 [: mid]
right_list = list1 [mid :]
mergesort (left_list)
mergesort (right_list).
base case

Penjelasan code ttg ‘Merge’