

$arr2LL([10, 20, 30, 40])$

$10 \rightarrow 20 \rightarrow 30 \rightarrow 40 \rightarrow$

$index = 0$

$current_node = 10$

$current_node.next = arr2LL(arr, index + 1)$

$0 + 1 = 1$

$index = 1$

$current_node = 20$

$current_node.next = arr2LL(arr, index + 1)$

$1 + 1 = 2$

$index = 2$

$current_node = 30$

$current_node.next = arr2LL(arr, index + 1)$

$2 + 1 = 3$

$index = 3$

$current_node = 40$

$current_node.next = arr2LL(arr, index + 1)$

$3 + 1 = 4$

$index = 4$

Base case \rightarrow return

function call order

$$\text{arrSum}([7, 2, 5, 0, 3])$$

17 → Final output

$$a[0] + \text{arrSum}(\text{arr}[1:])$$

$$7 + \text{arrSum}([2, 5, 0, 3]) = 10$$

$$a[0] + \text{arrSum}(\text{arr}[2:])$$

$$2 + \text{arrSum}([5, 0, 3]) = 8$$

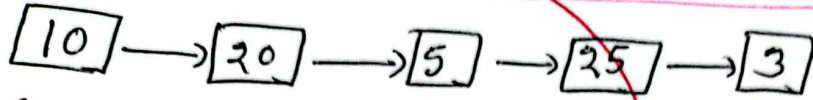
$$5 + \text{arrSum}([0, 3]) = 3$$

$$0 + \text{arrSum}([3]) = 3$$

length == 1 (Base Case)
Return $a[0] = 3$
 $= a[0]$
 $= 3$

function returning value order

$\text{find_max}(\text{head}) \rightarrow 25 \rightarrow \text{Final output} = 25$

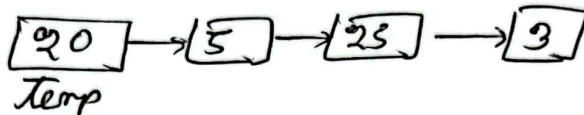


~~head~~
Temp

max = 10

max-node-rest = $\text{find_max}(\text{temp.next}) \rightarrow 25$

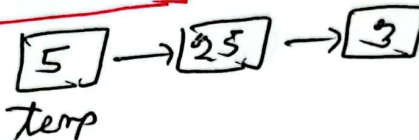
return $10 < 25 \therefore \text{max} = 25$



max = 20

max-node-rest = $\text{find_max}(\text{temp.next}) \rightarrow 25$

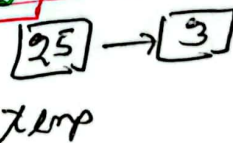
return $20 < 25 \therefore \text{max} = 25$



max = 5

max-node-rest = $\text{find_max}(\text{temp.next}) \rightarrow 25$

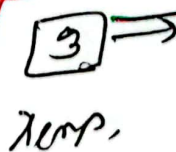
return $5 < 25 \therefore \text{max} = 25$



max = 25

max-node-rest = $\text{find_max}(\text{temp.next}) \rightarrow 3$

return $25 > 3 \therefore \text{max} = 25$



Base case
temp.next is none.
return value (3)