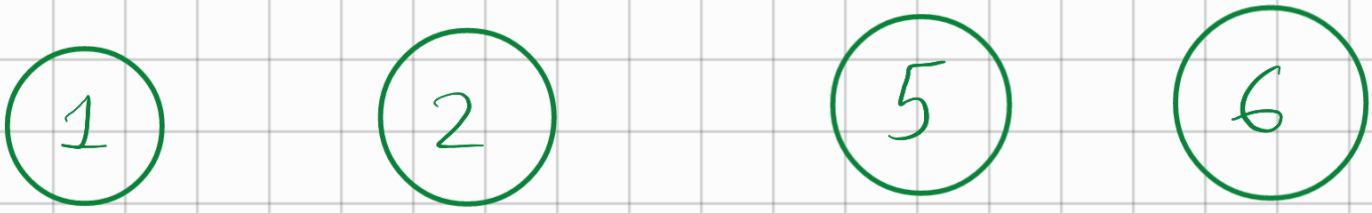
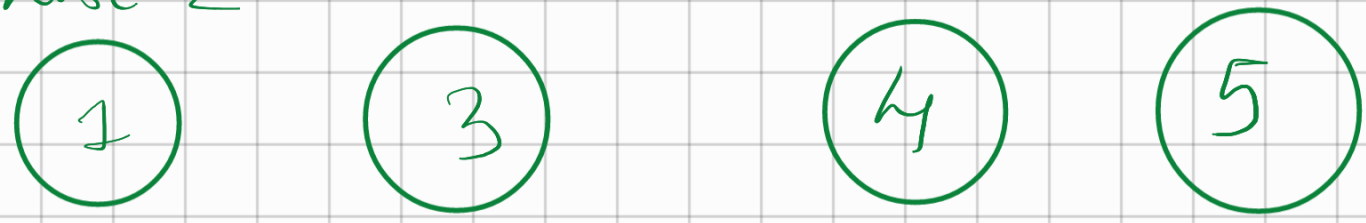
Merge two sorted lists.

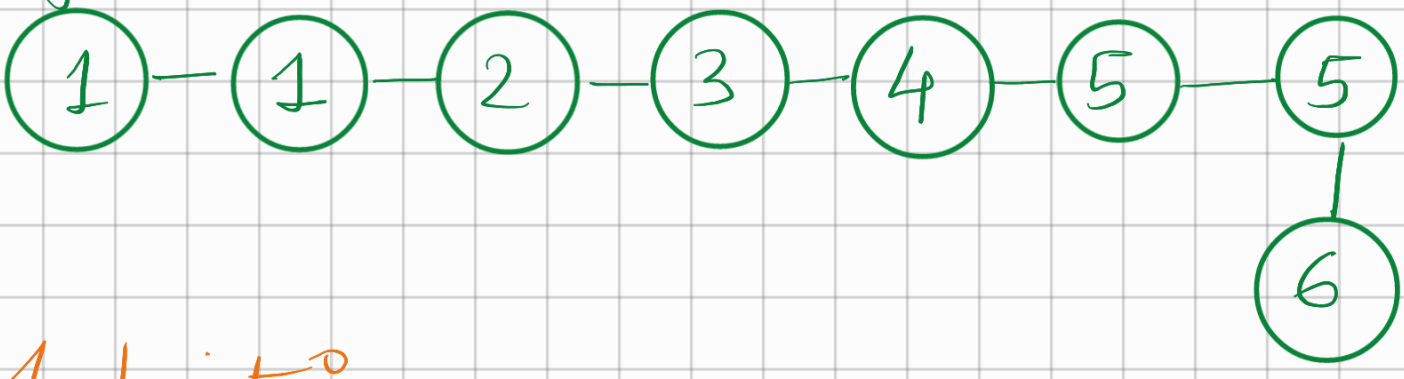
List 1



List 2



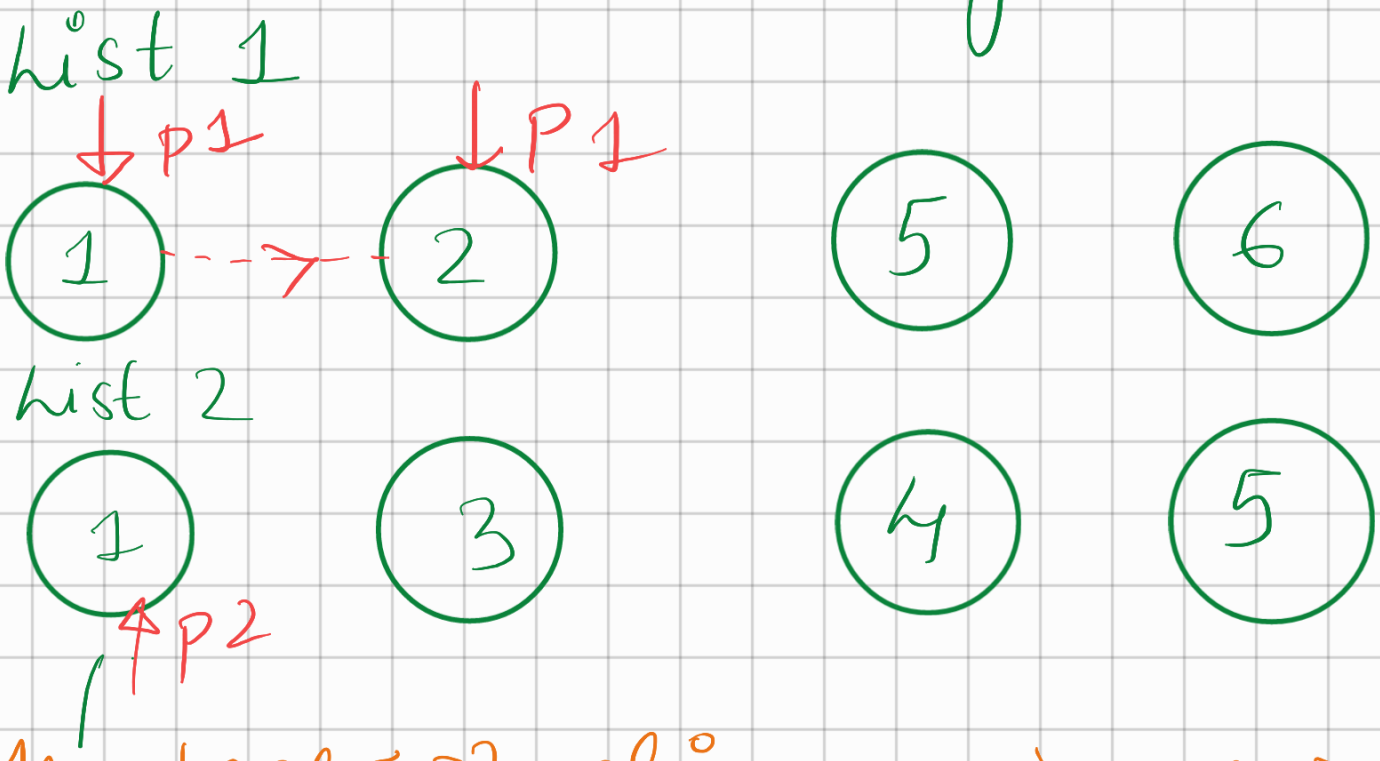
final Answer.



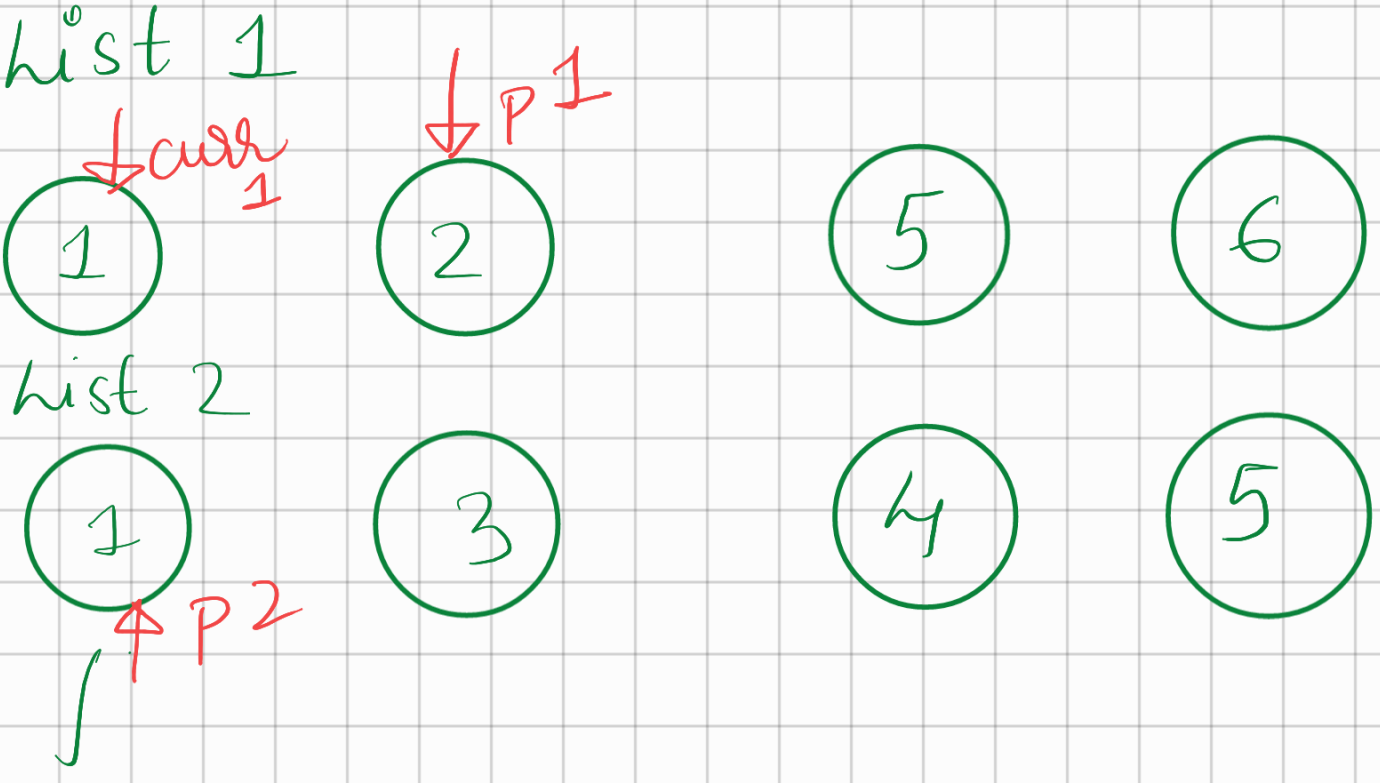
Intuition

- We require two pointers for moving through list 1 & list 2 separately.
- Since comparison logic is same & the value which is lesser is set to prev value's next the entire logic can be put in a loop / Recursion

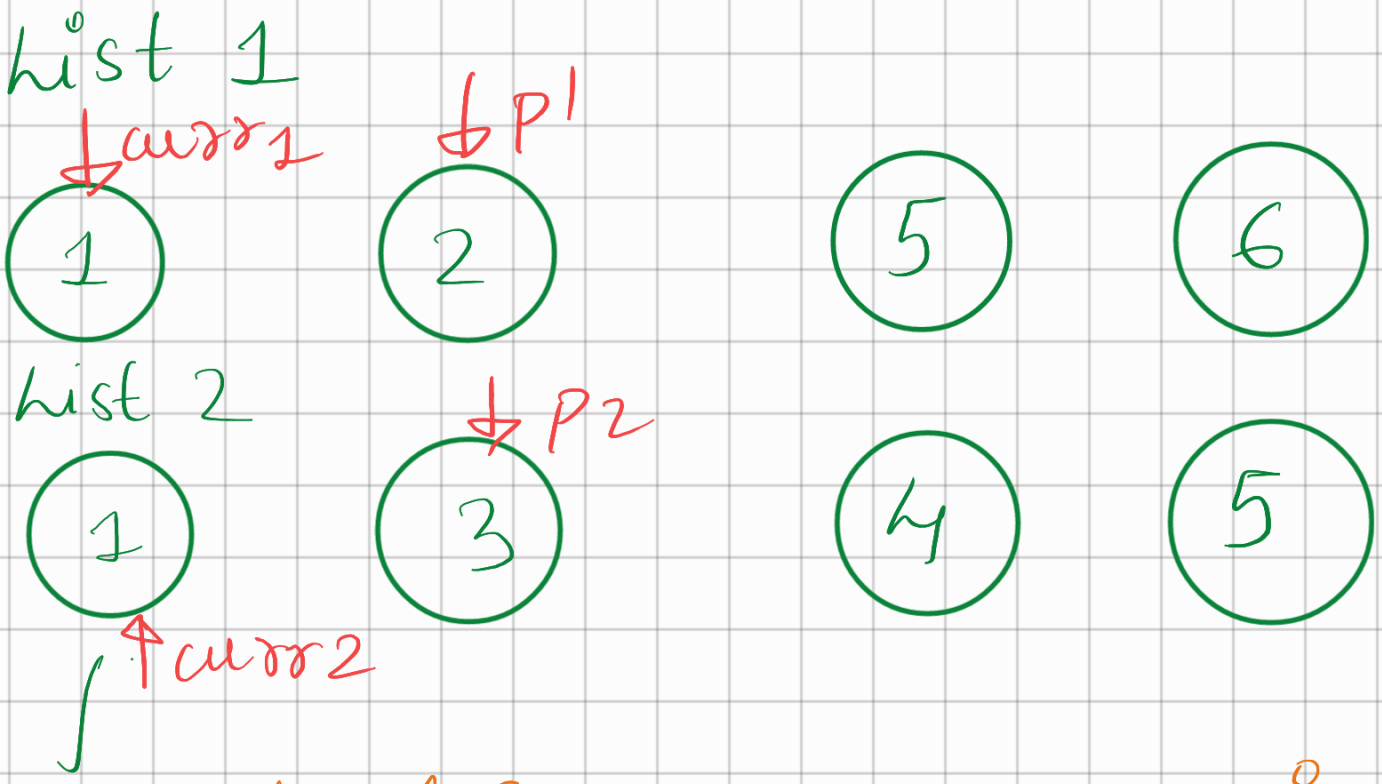
for Above Example Dry Run.



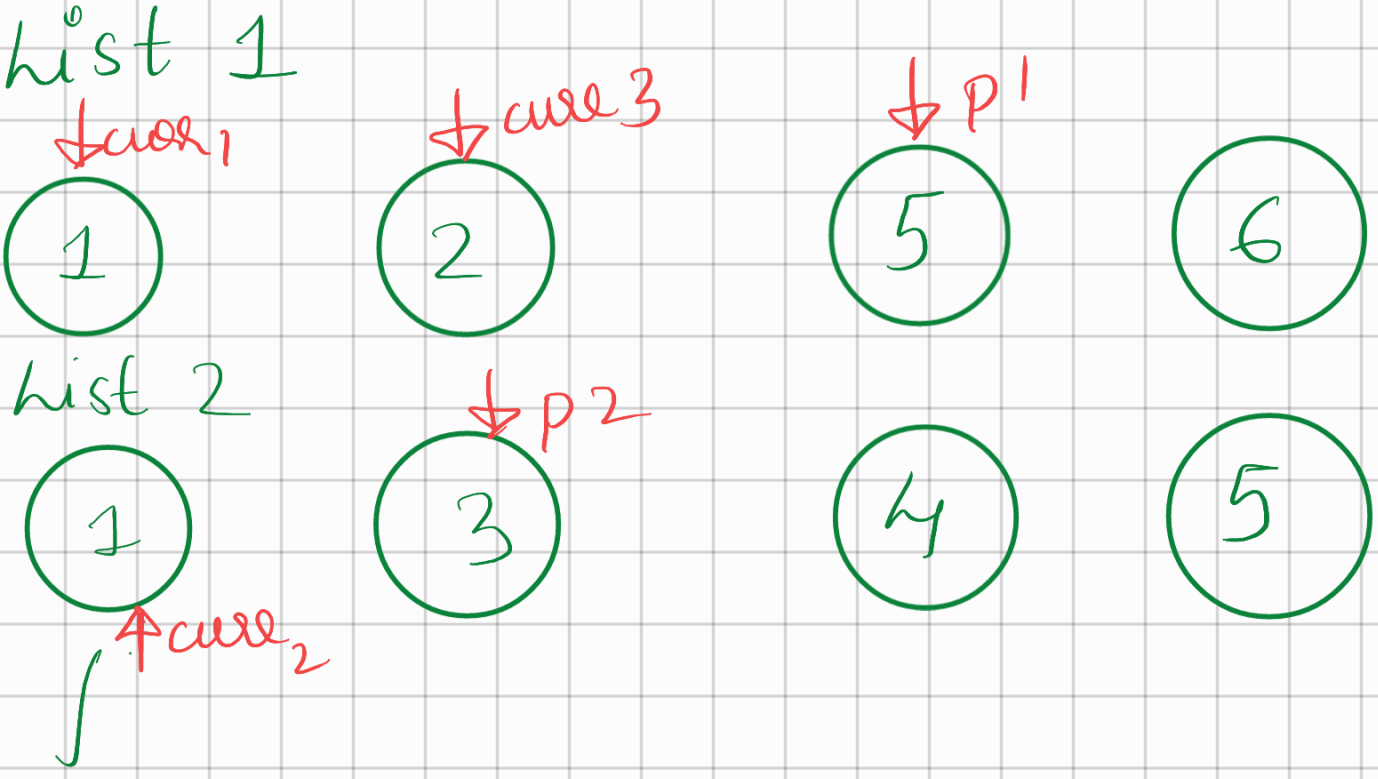
As p1.val ≤ p2.val; we assign current to p1 & increment p1. & pass the updated values to the function & assign it to current when the function returns its values in future.



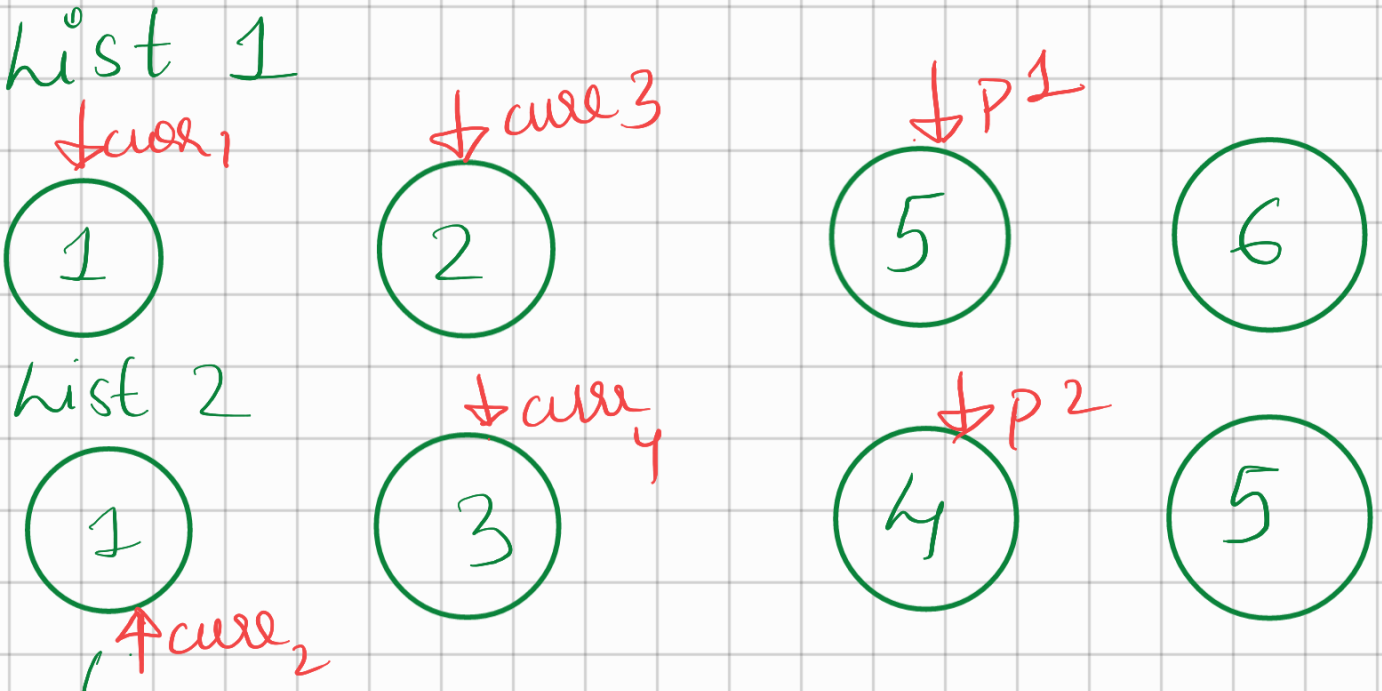
Now p1.val > p2.val so p2 is assigned as current & p2 is incremented.



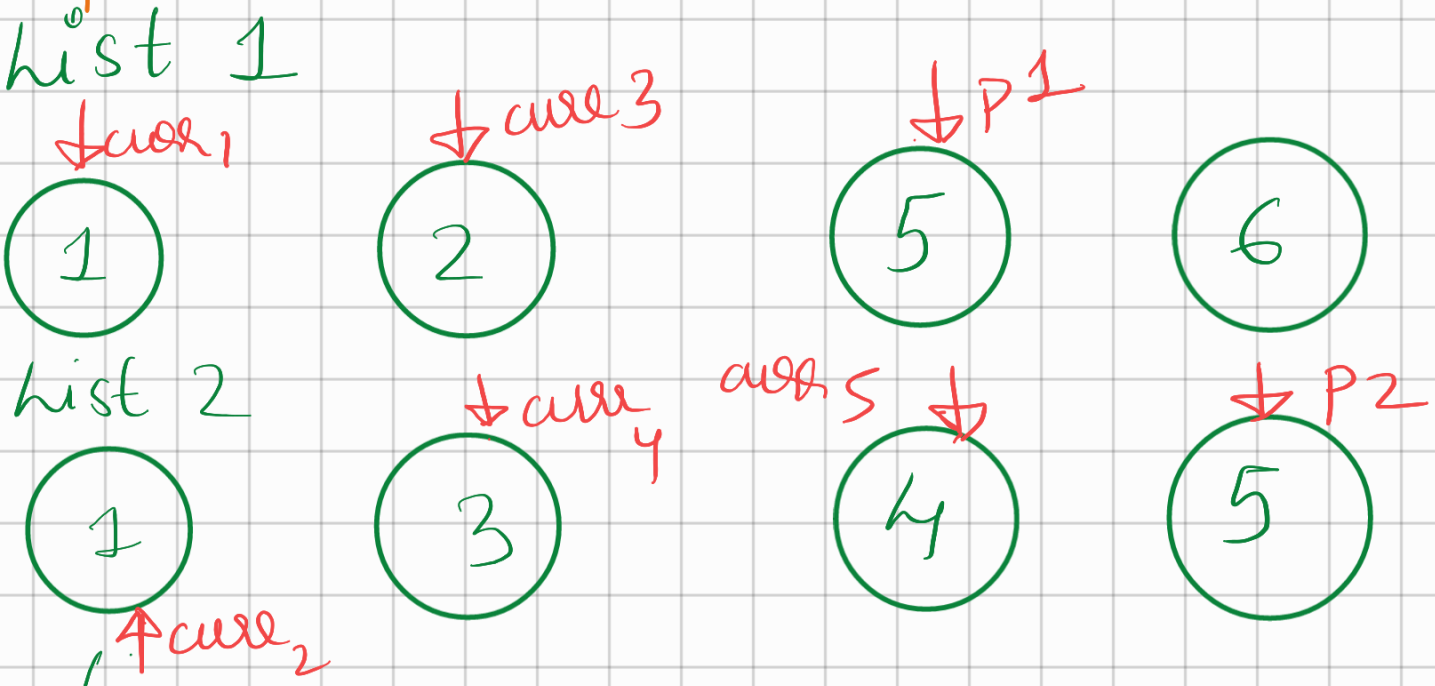
Now p1.val ≤ p2.val so curr is changed to p1 & p1 is incremented.



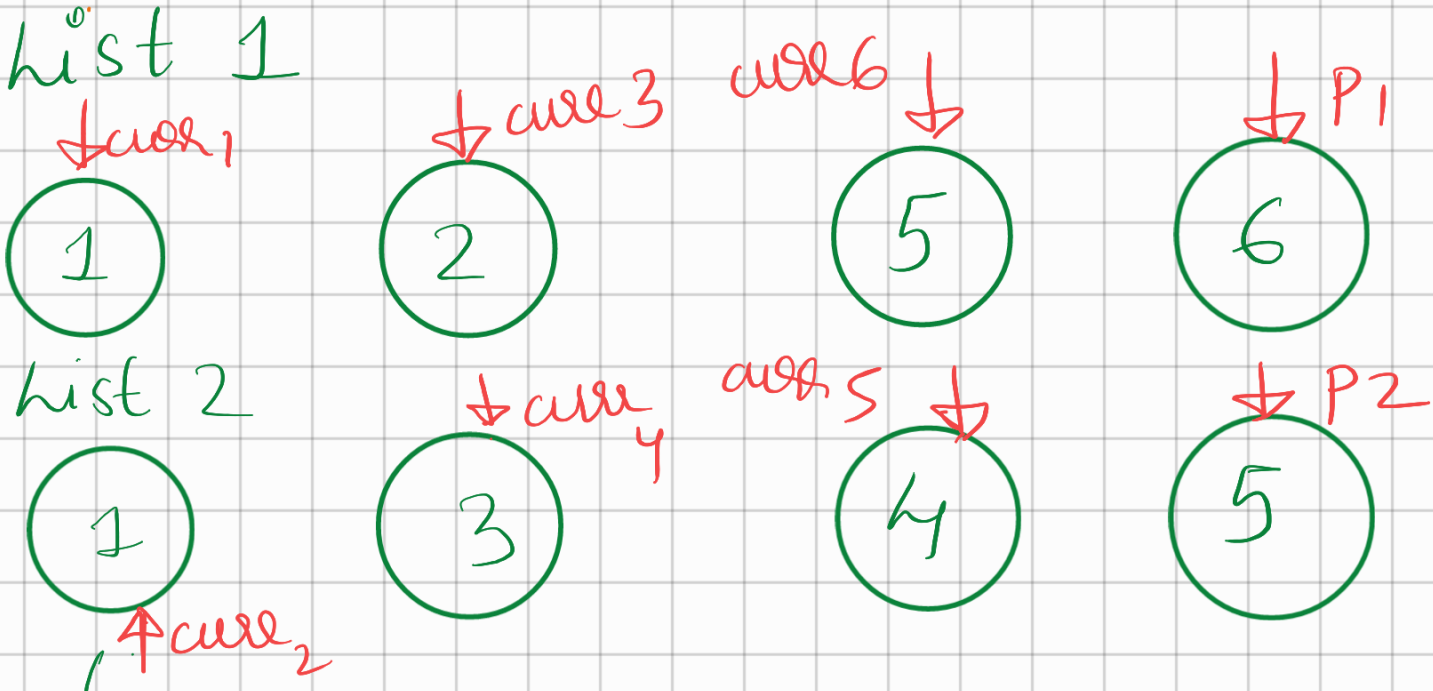
Now p1.val > p2.val so curr is updated to p2 & p2 is incremented.



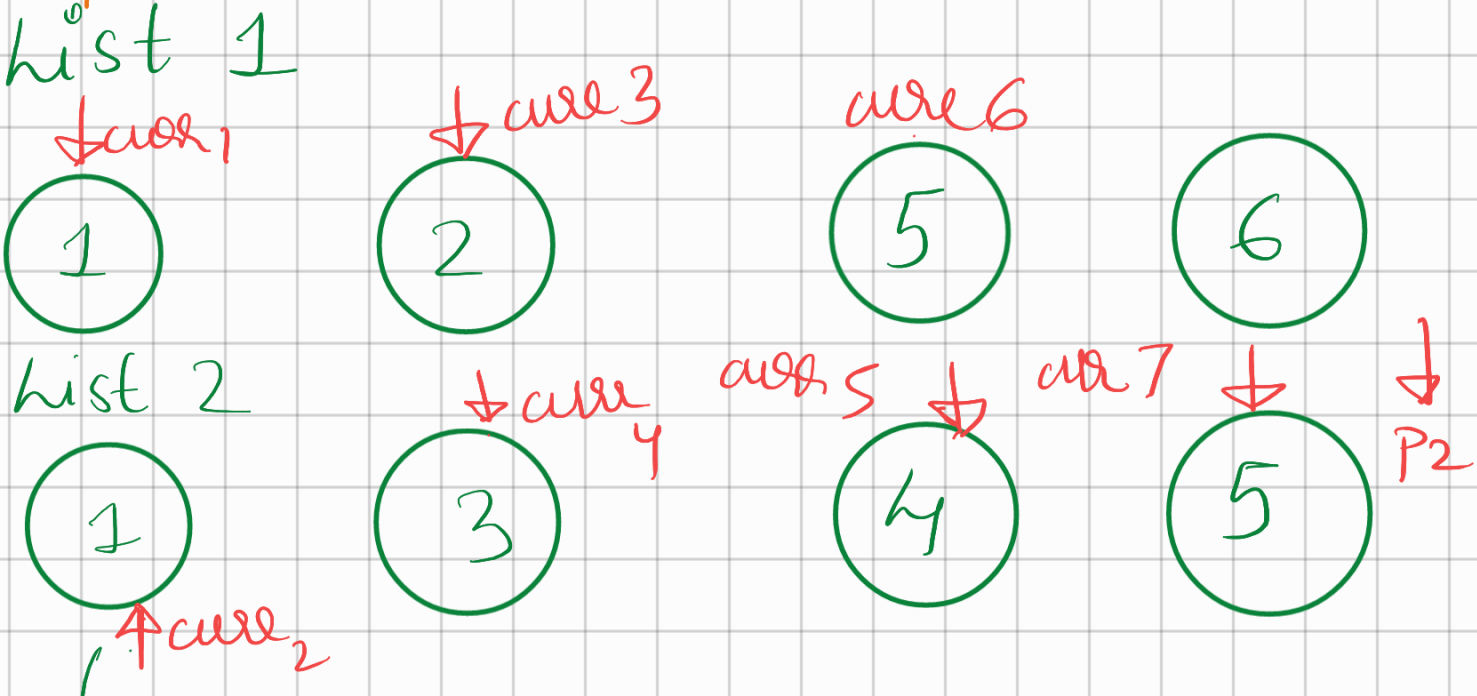
Now p1.val > p2.val so curr is updated & also p2



Now p1.val ≤ p2.val so curr is updated & p1 incremented.



Now p1.val > p2.val so, curr is updated & p2 incremented up



Now since p2 is null so p1 as a whole should be returned & should be assigned to curr.next

hist of curr's values have been drawn wow Recursion would close itself. as :-

cur7.next = p1 {cur = 7}

cur6.next = cur7 {cur = 6}

cur5.next = cur6 {cur = 5}

⋮

cur1.next = cur2 {cur 1}

& at the end current 1 is returned as the head.

TC = O(m + n)

SC = O(m + n)

Solve iteratively to decrease space to O(1)