Linked List:



Node having two part data and next.

Always Last node with next = None

First node as well header equal to None.

First node = header

Node structure = data©, next©

Inkedin structure = head©, internally create object of Node

Adding at end : self.head constant mean after addition no need to add anyting else in the node

Adding at beginning: simple iteration and add at beginning of sefl.head

Exceptional for ramove at index 0:

Self.head = self.head.next

Always remember to handle all the cases like w



Binary search : sorted list O(logn)

Key element to remember : def binary\_search(ls, n, left, right):

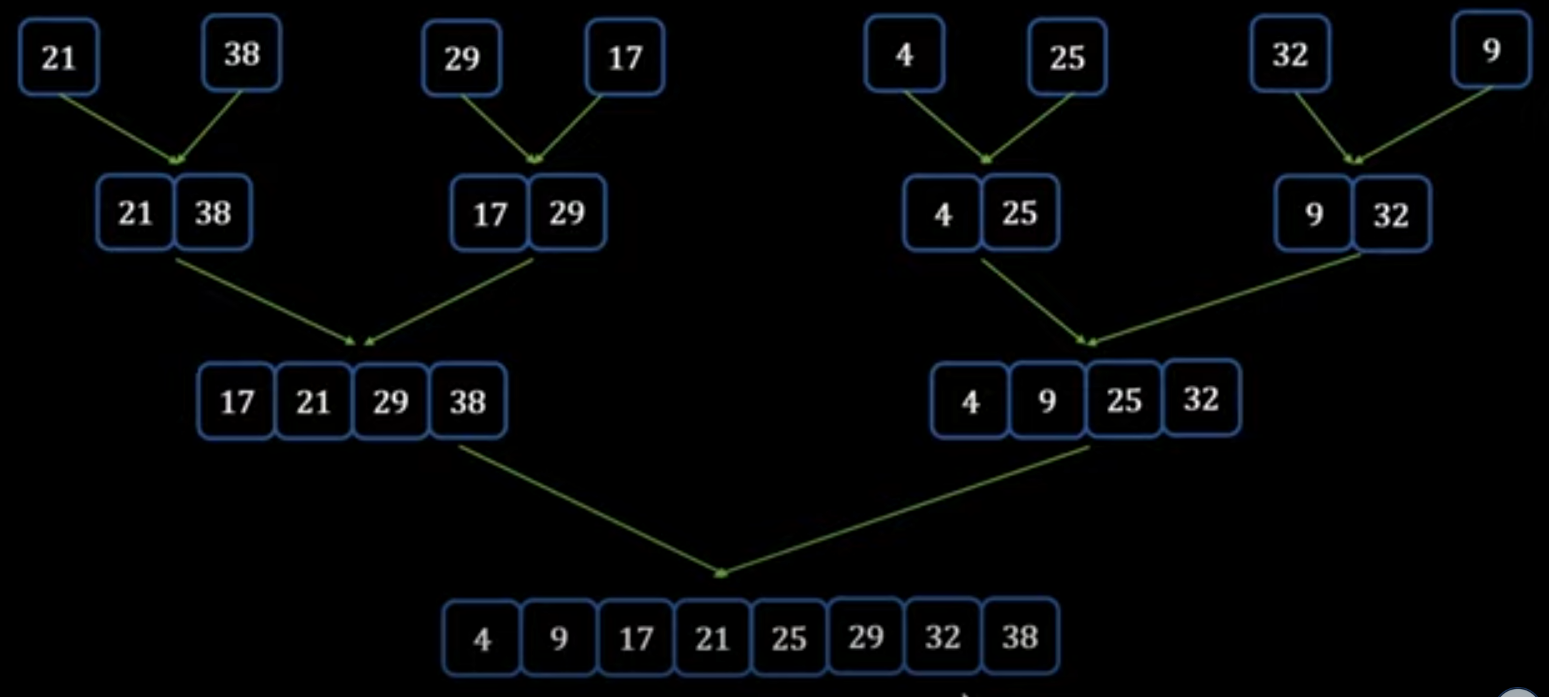
# Bubble sort

Sort by pair wise first center two elements all along the list then take sorted list element and similar ways do two sorting again until sort all.



# Merge sort

Split list in to multiple smaller left and right list. Sort and merge smaller group into main array by maintaining the pointer.



# Insert sort

Maintain same list and sort by maintaining pointer of list one by looping over the list.

