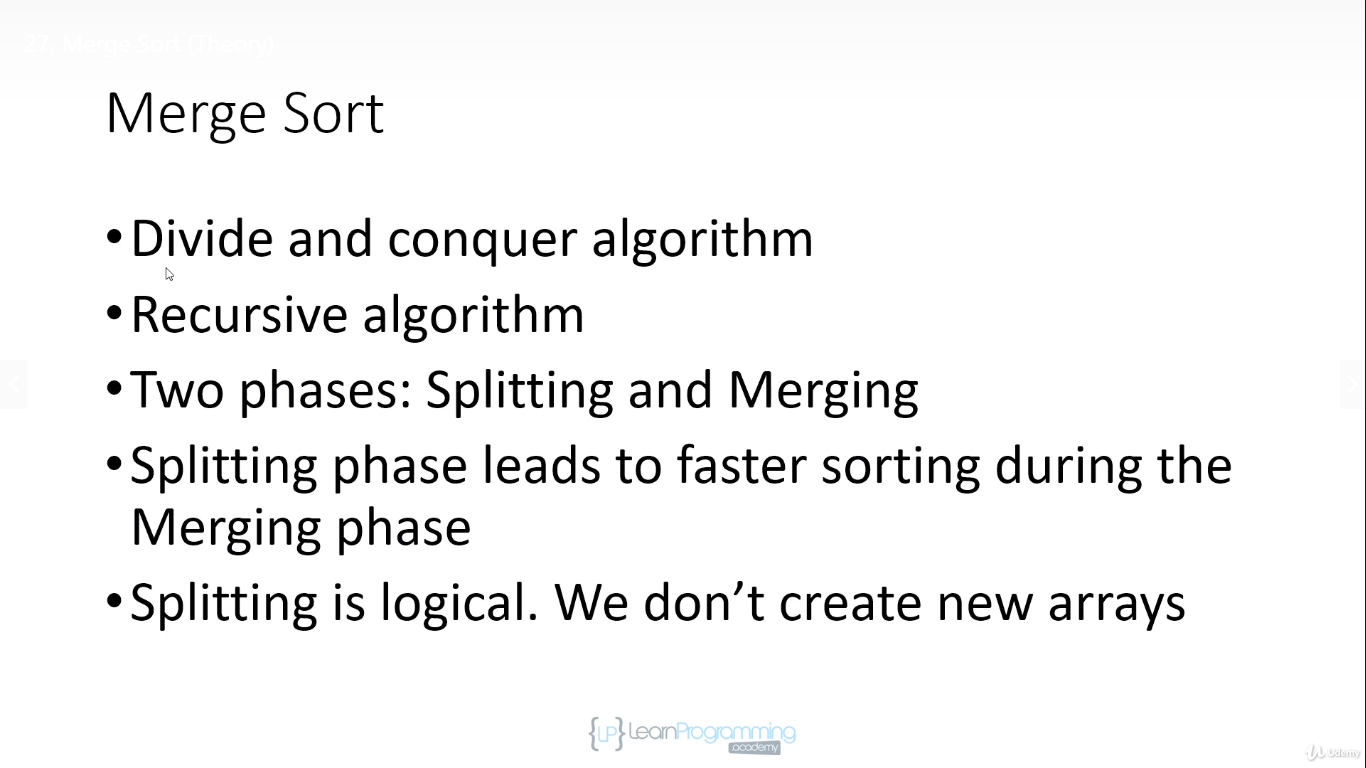
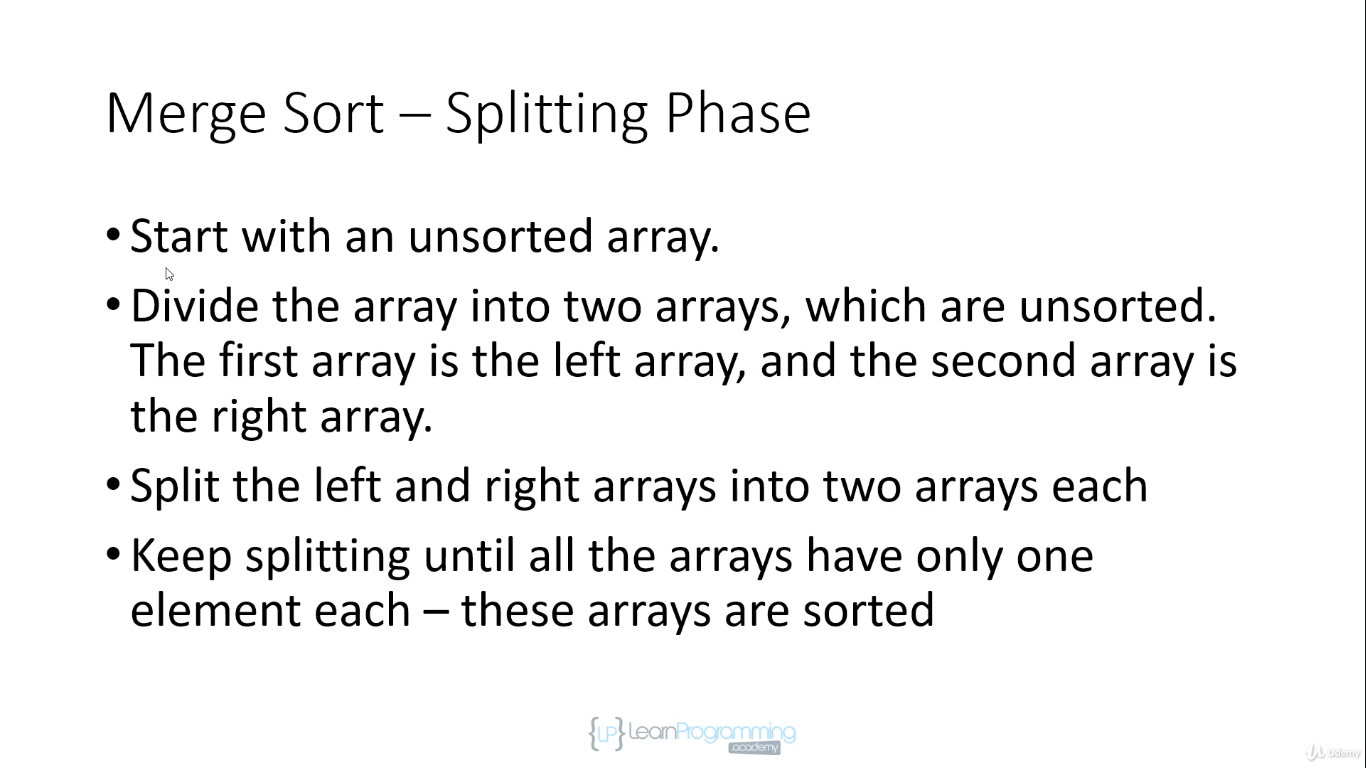
**Merge Sort**

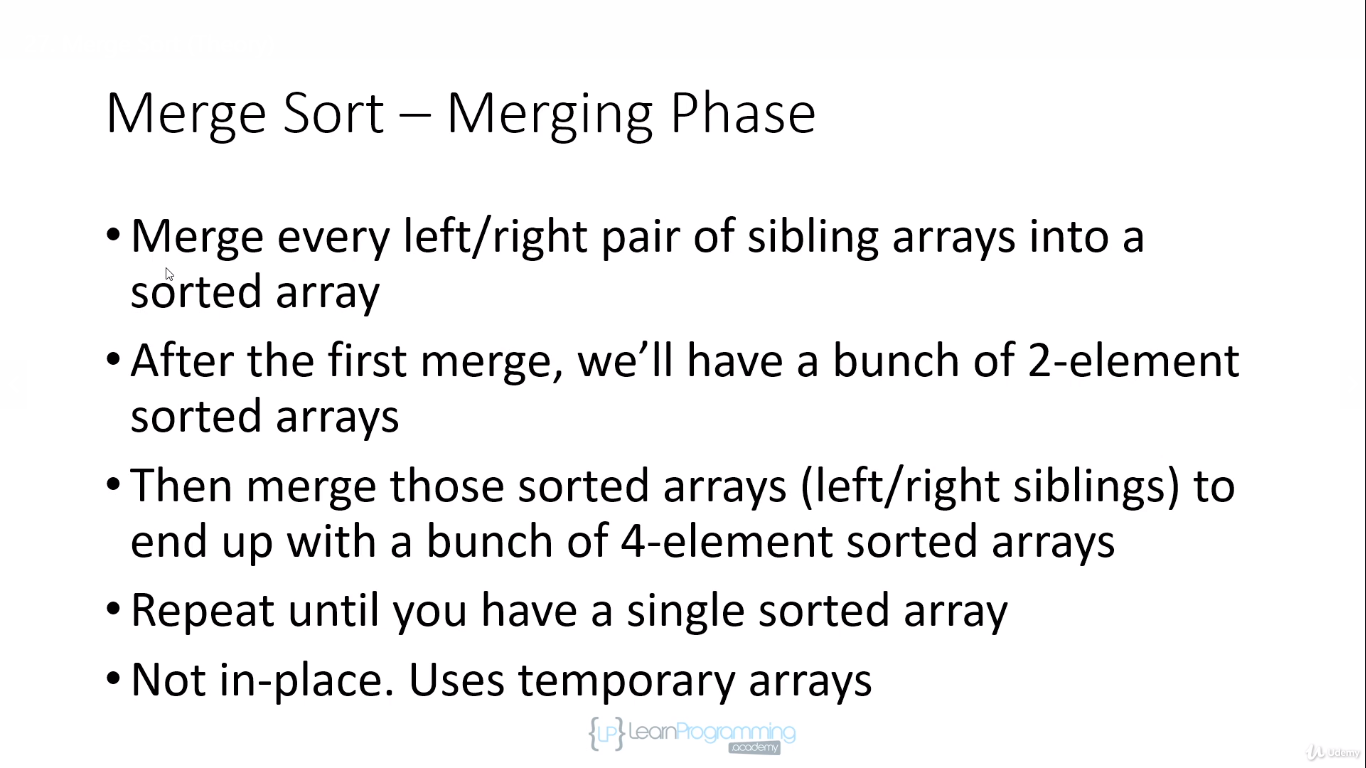


We use indices to keep where arrays has been split

**Splitting phase**



**Merging Phase**

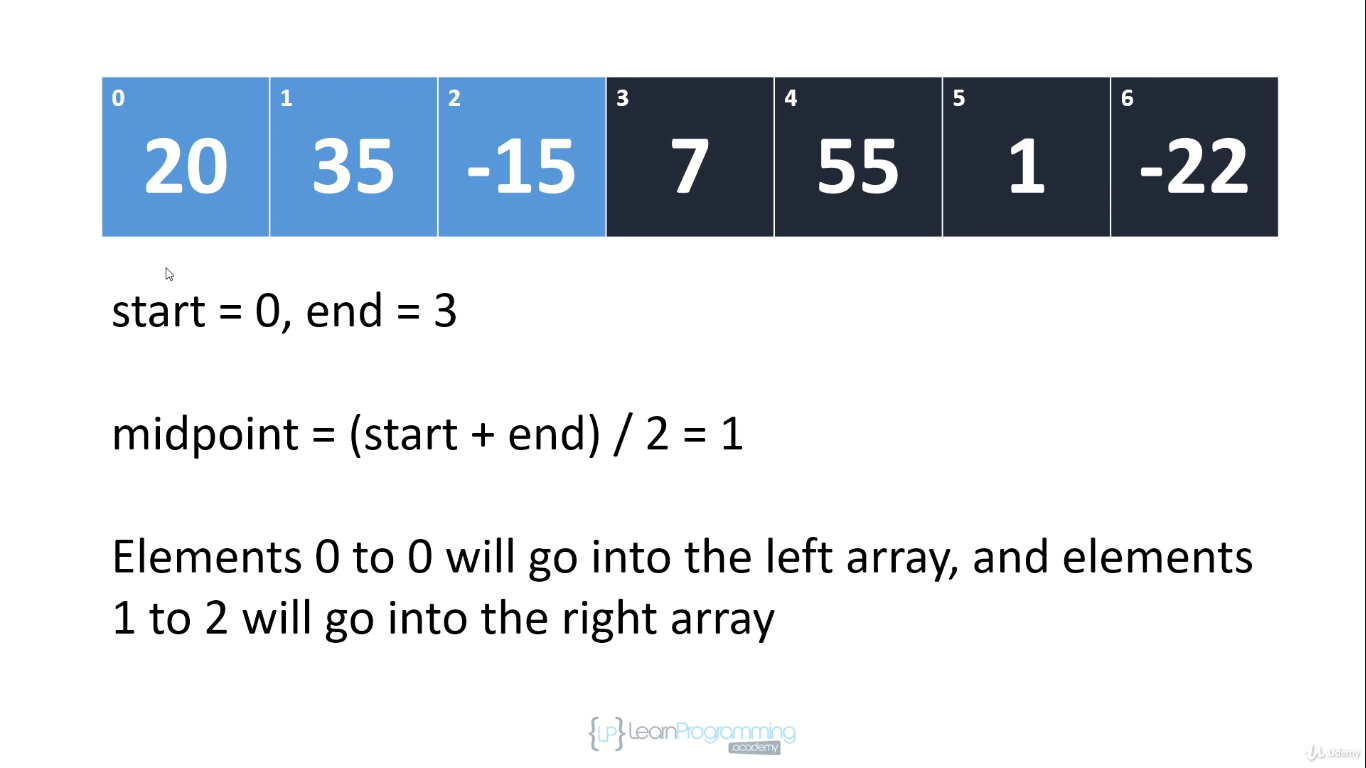


Every Resulting array will be sorted from merge.

Merging phase in not-in place as its use temporary arrays.

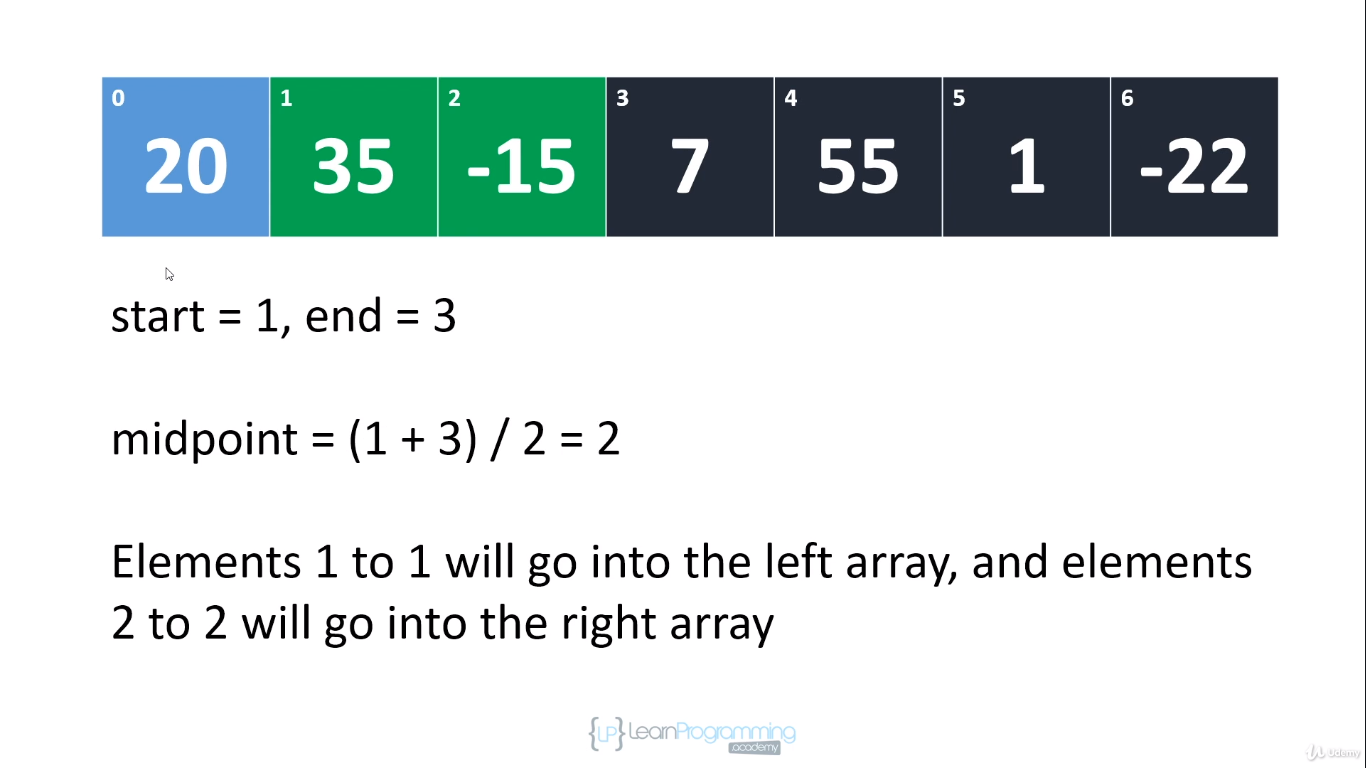


Left Array Split phase

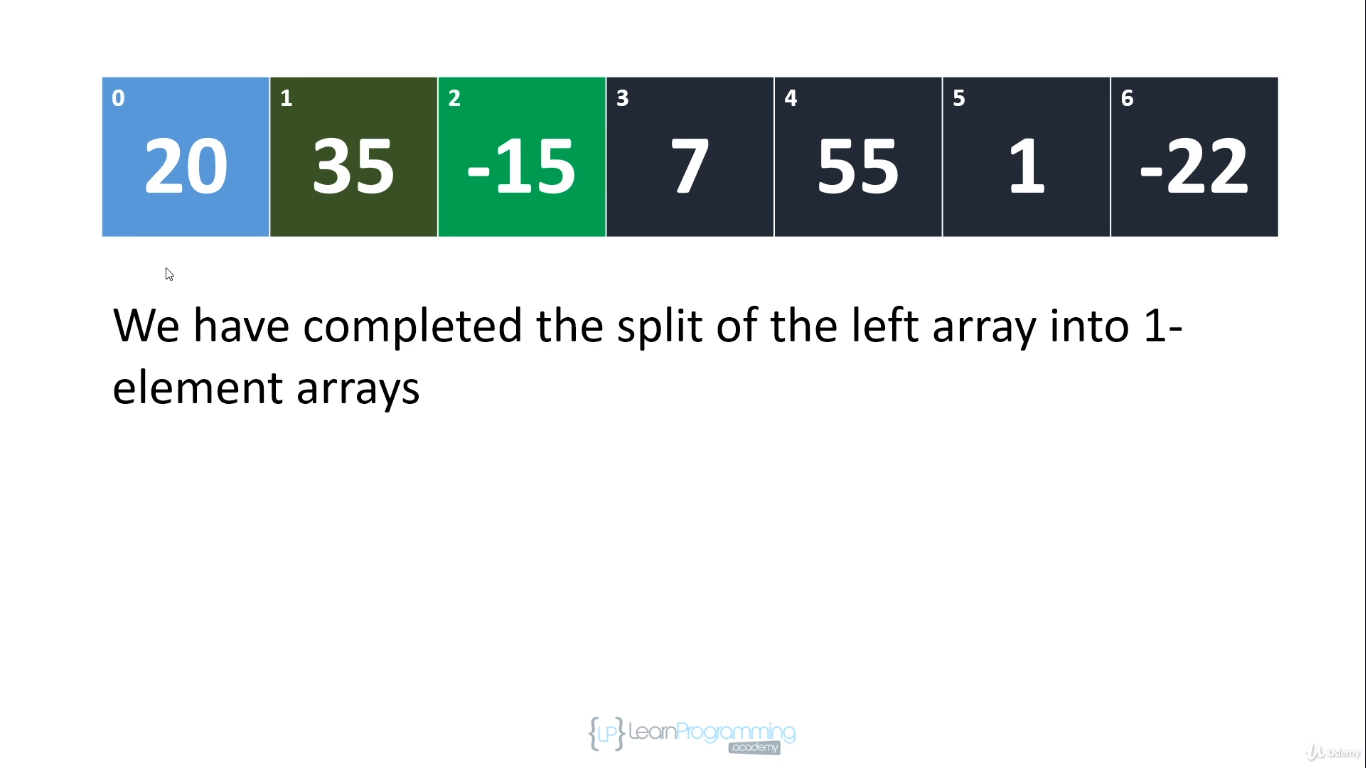


Left Array Split sub phase

End index is always one greater than the index of last element in the array



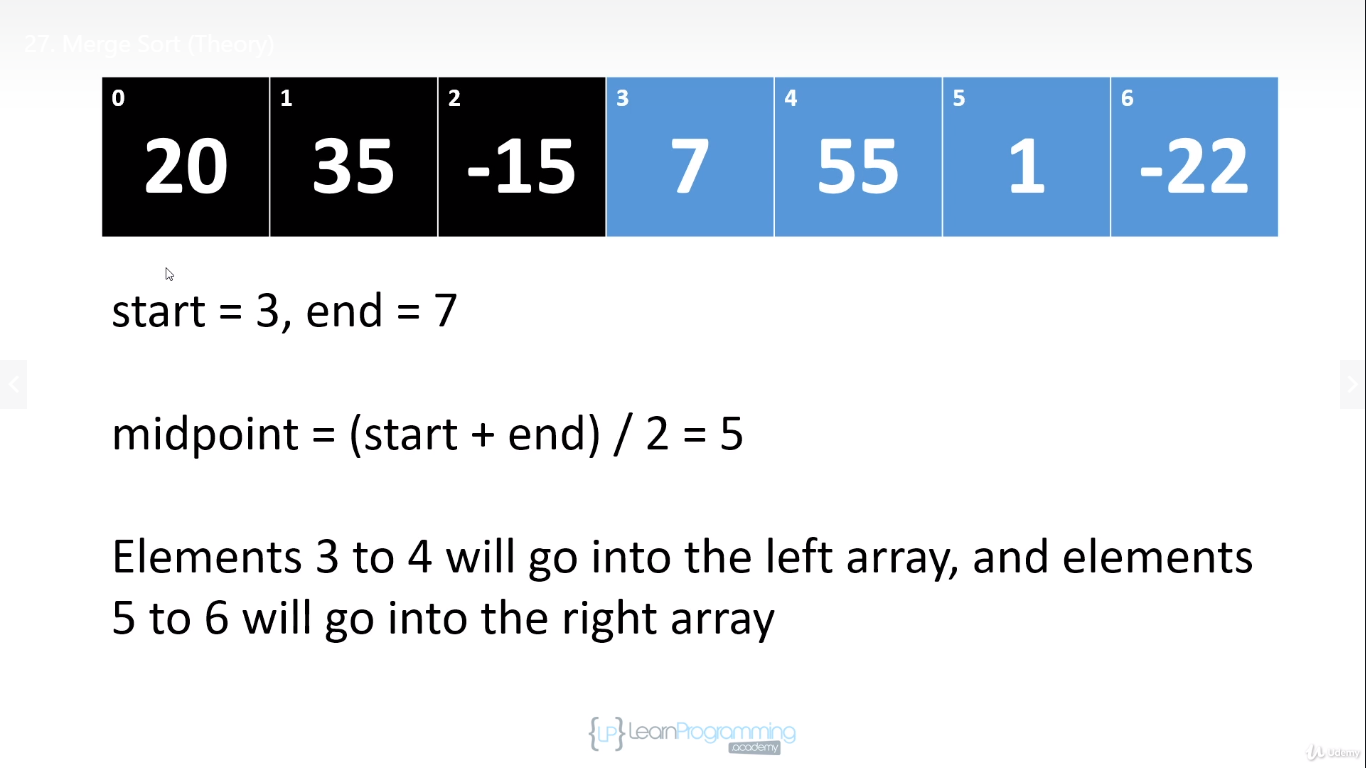
Split the right array of left sub array.



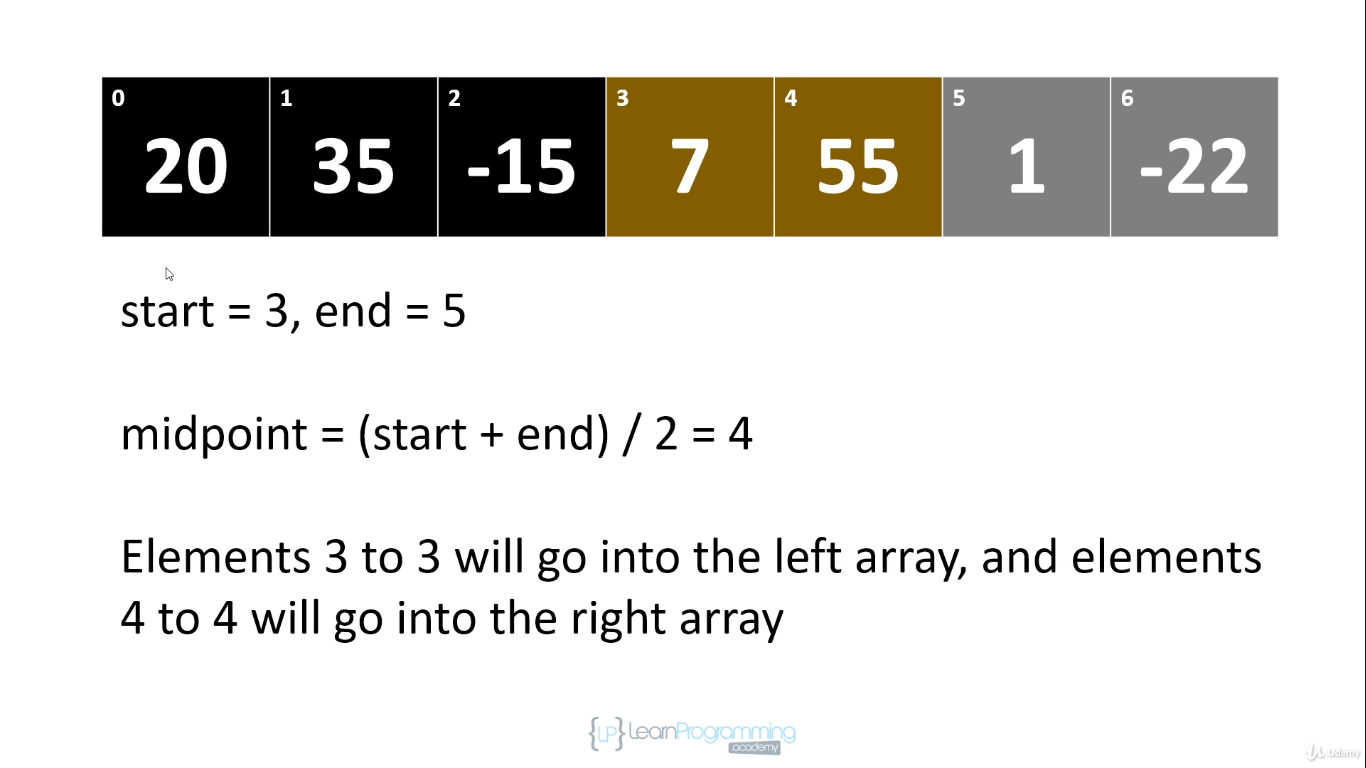
At the time of merging we will always merge the last splited array.

So the first merge we do is -15 and 35 on the left side after sorting and then after -15 35 merging completes then we merge them with 20 like this only we proceed.

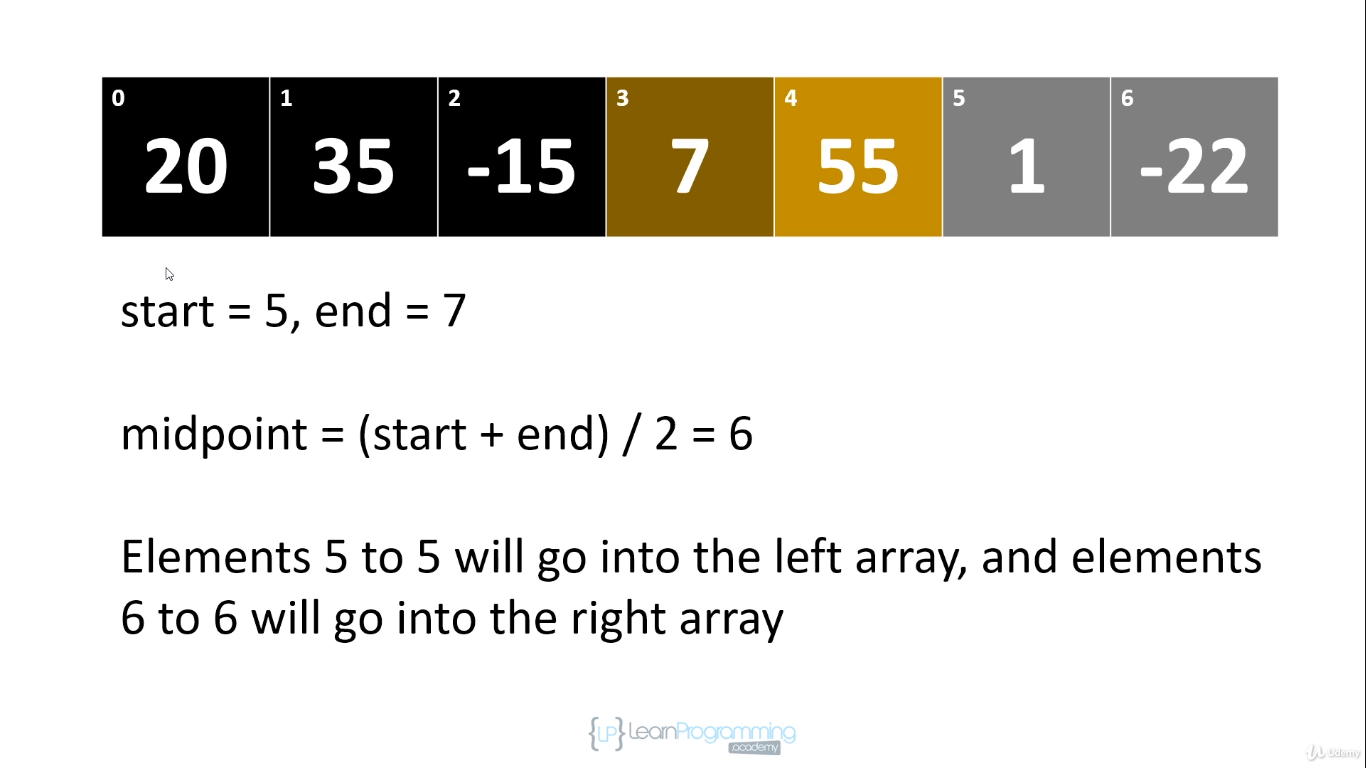
Top right array



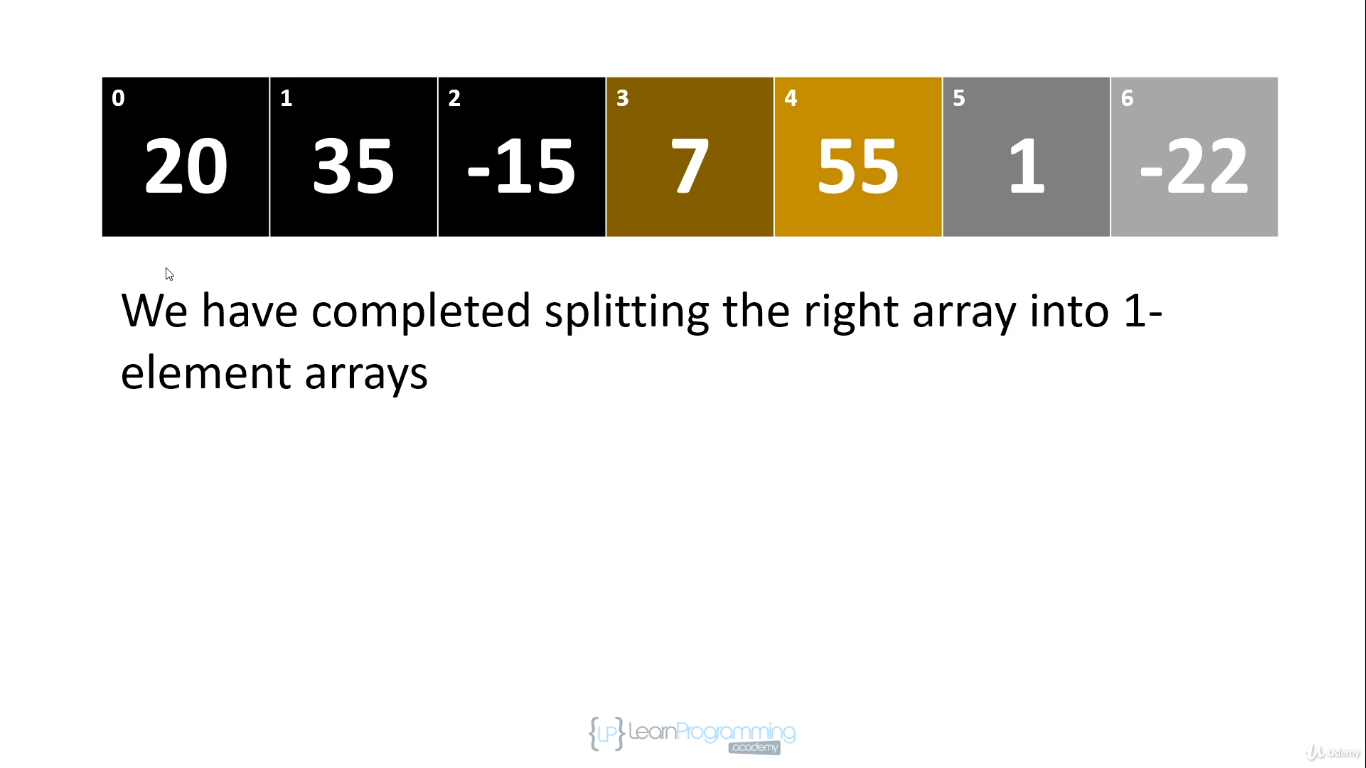
Now lets work on spliltting right array again



We have done split left array on right side

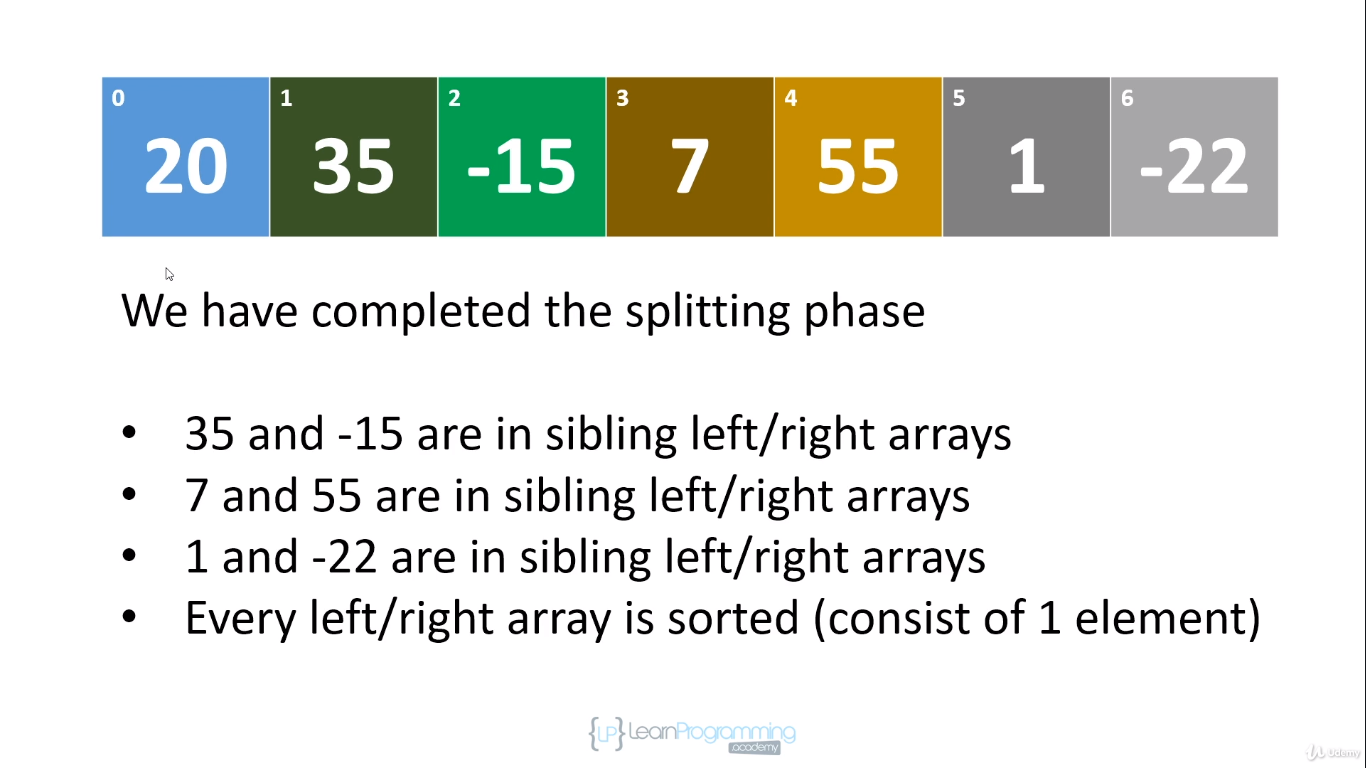
now we split right array on right side

now we have completed split on right side of right array

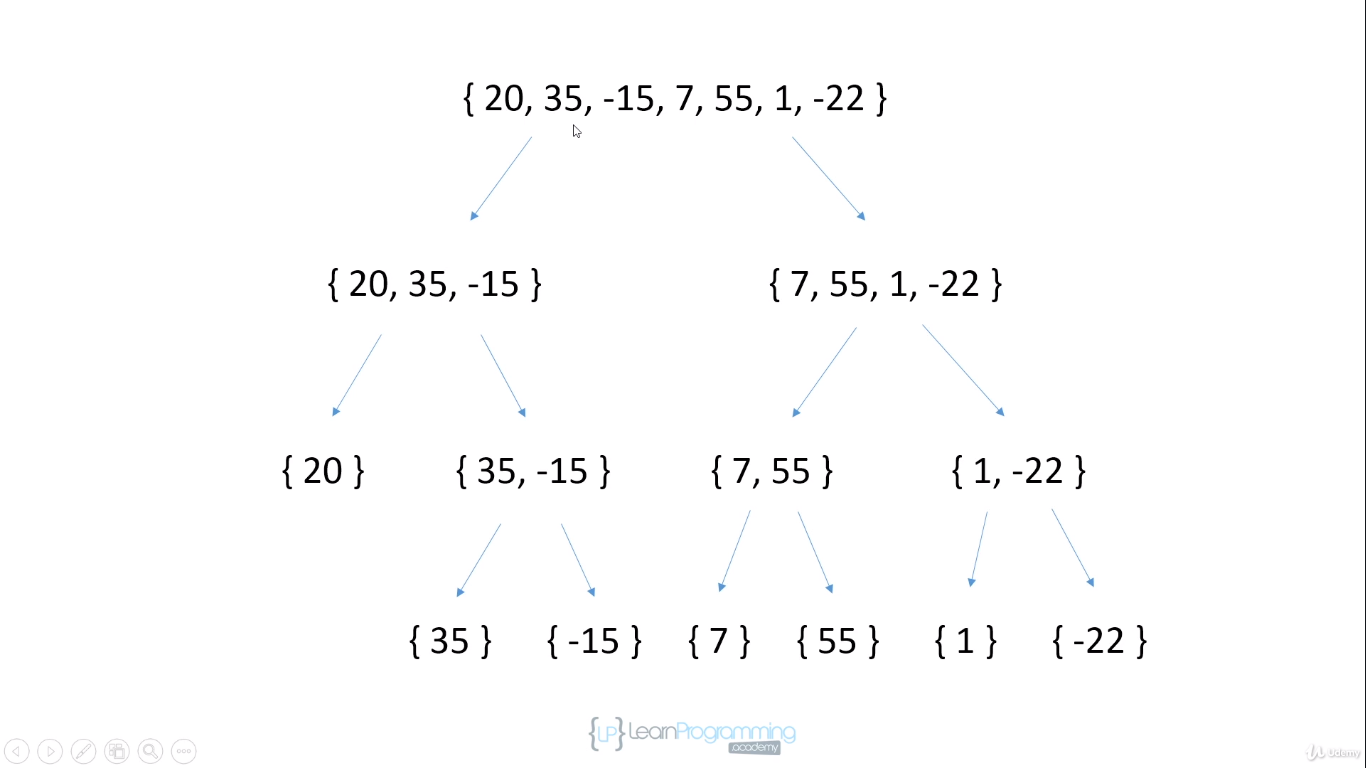


7 and 55 are sibling arrays so they will be merged first.

SO NOW WE MERGE AND Every time we merge we actually sort. So the resulting array after merging is sorted.

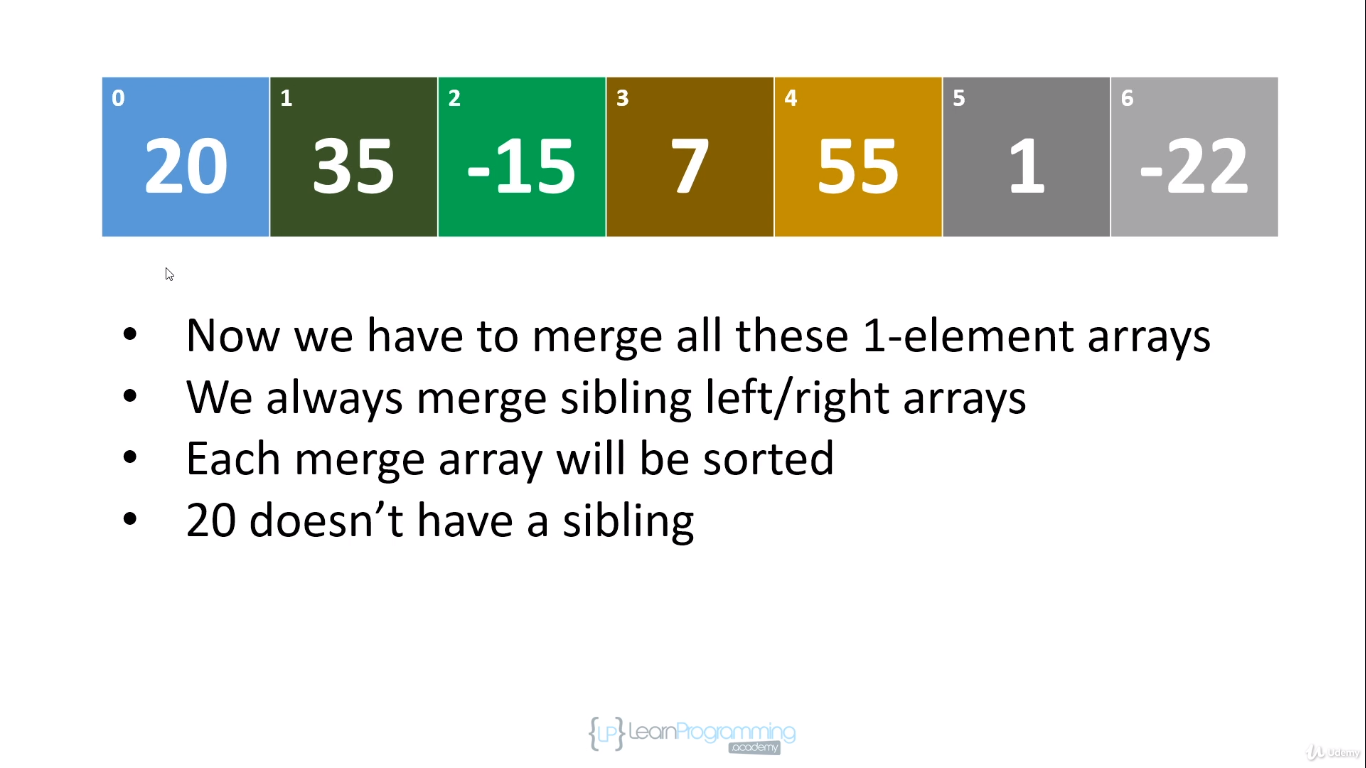


Started with our normal array and then splitting begins

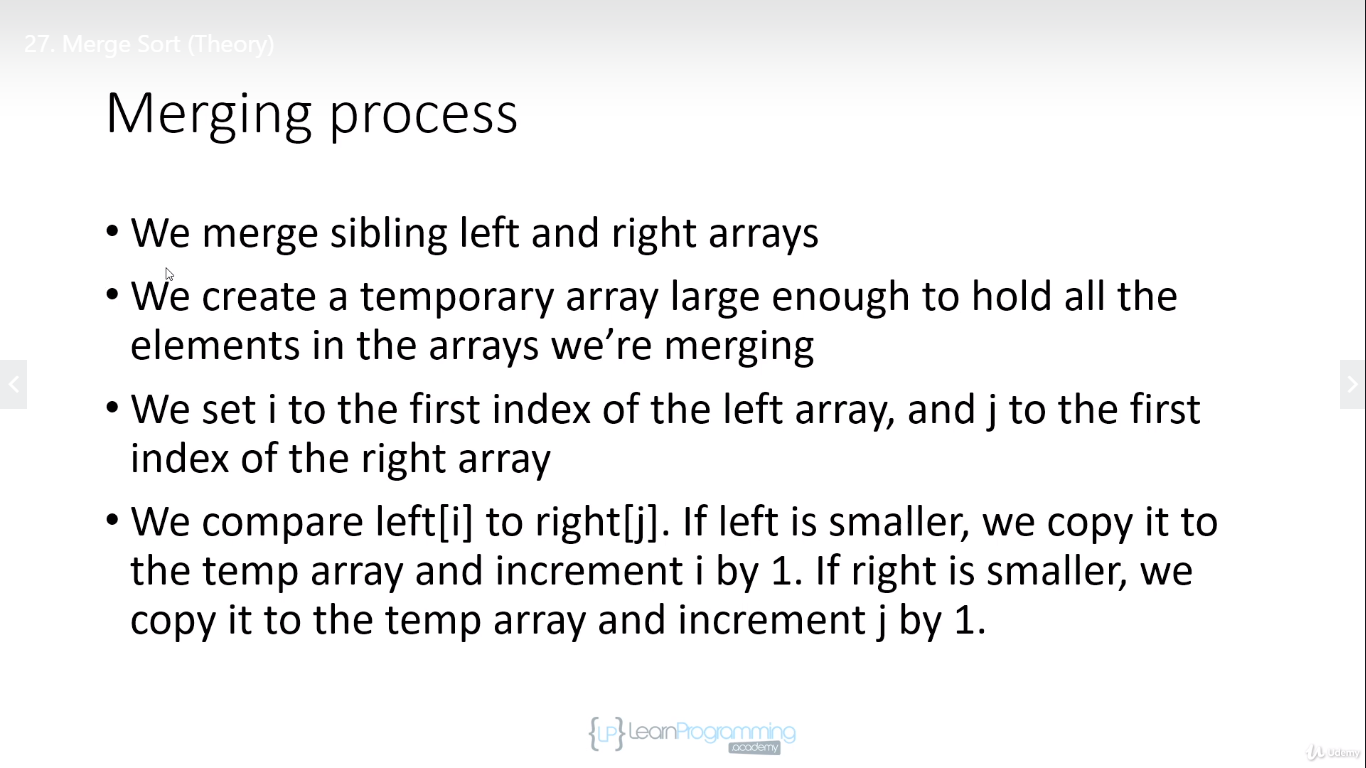


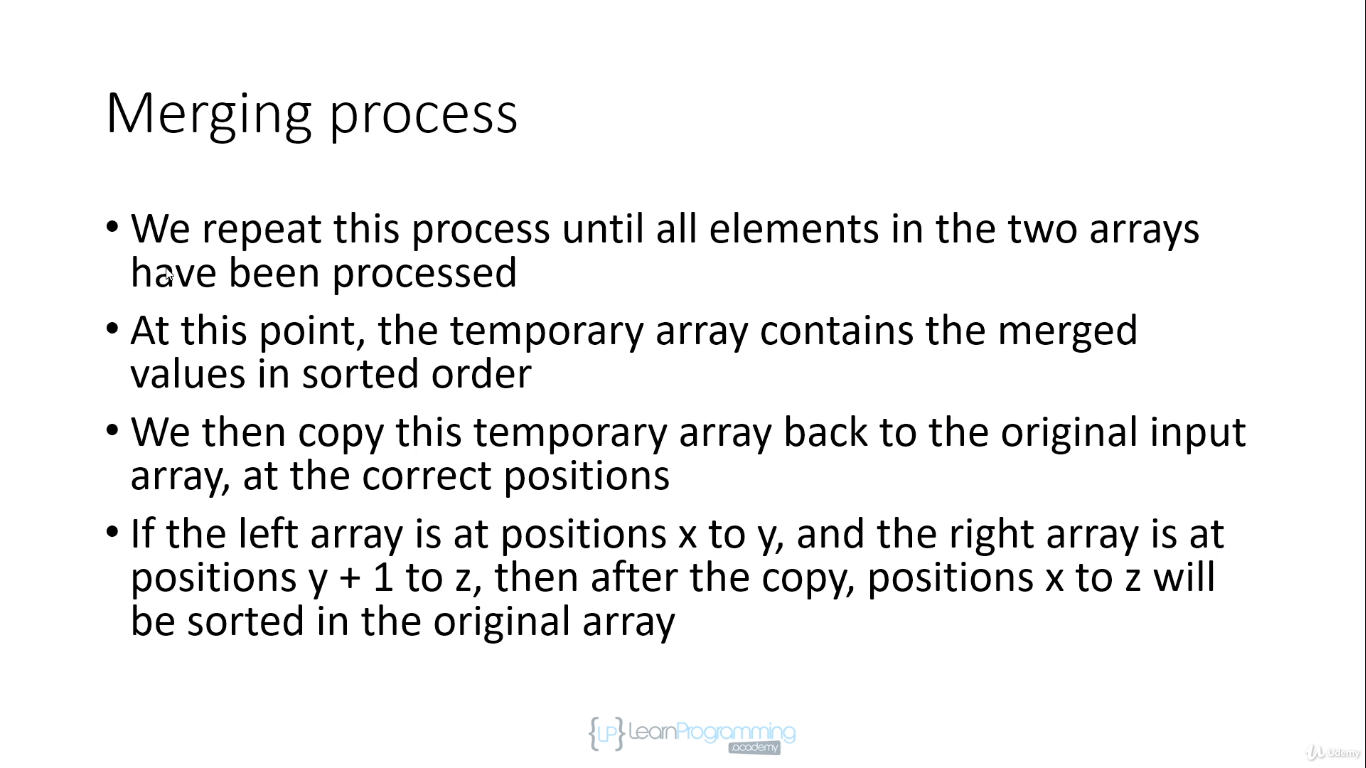
Because of the recursive nature of the implementation we are going to handle that entire left side of array before we work on right side of array.

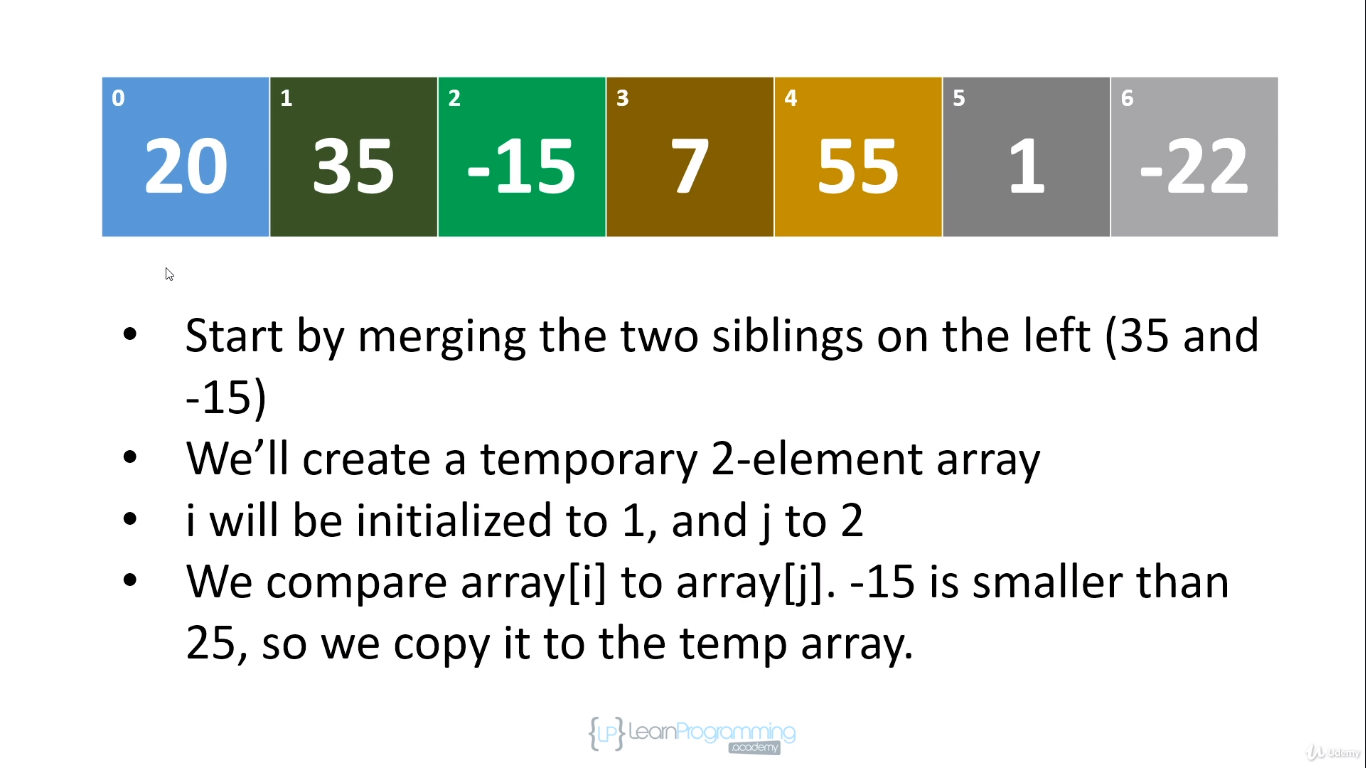
When we merge we go bottom up

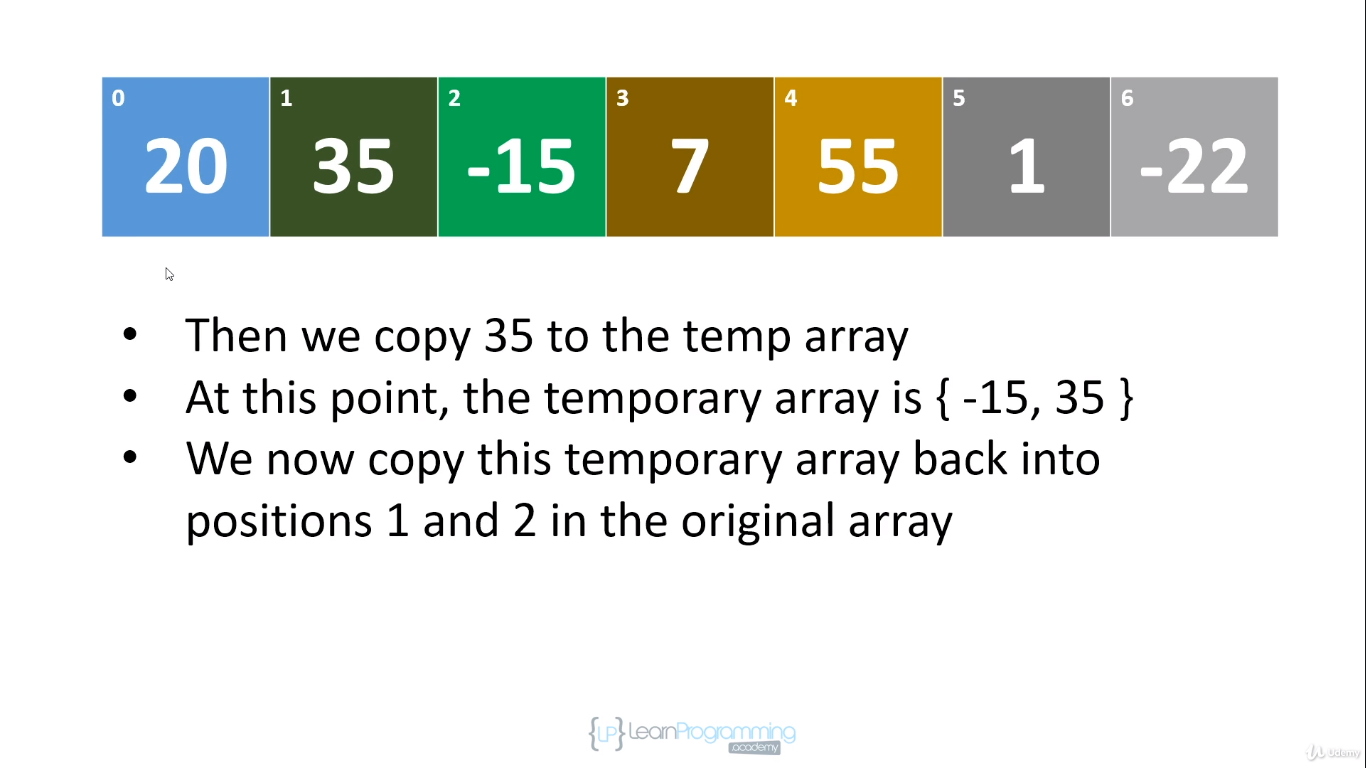


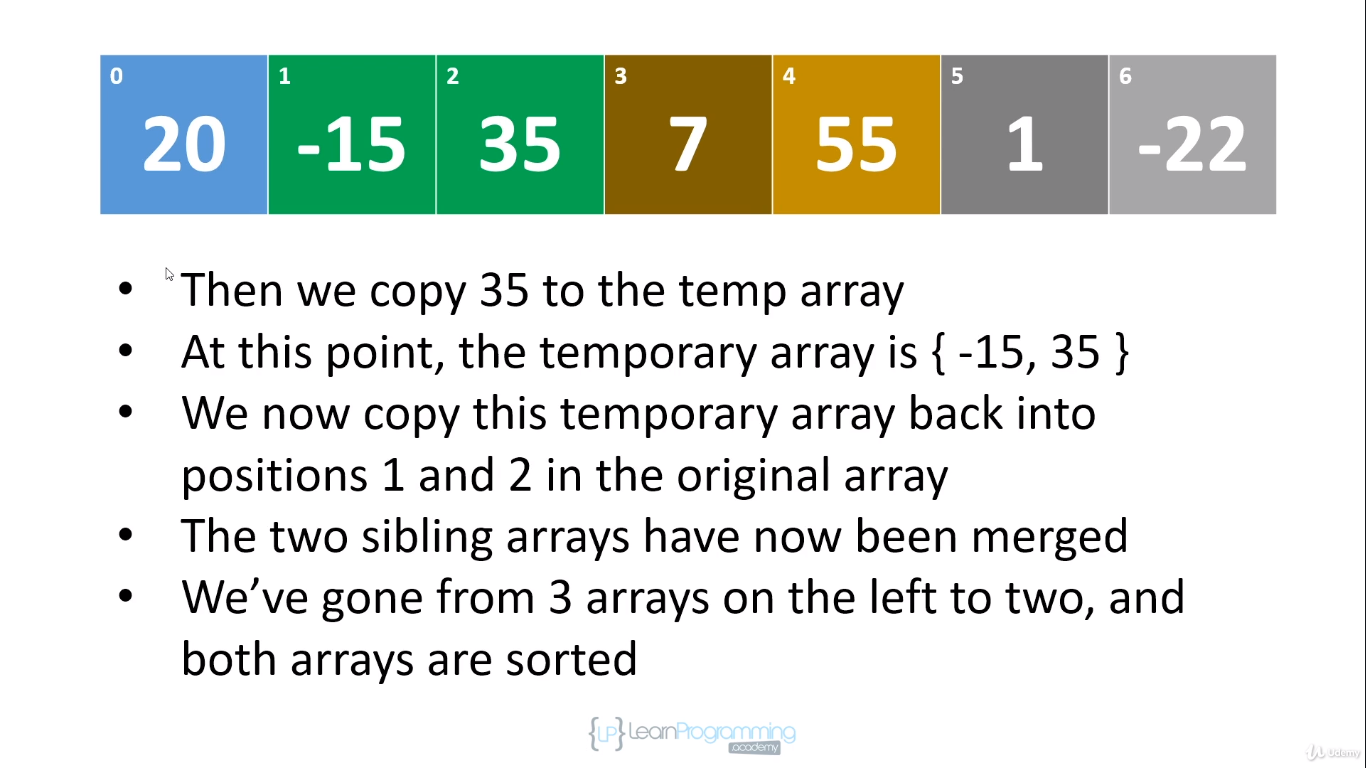
At first round our first temp array will be length 2 because we are going to merge two elements

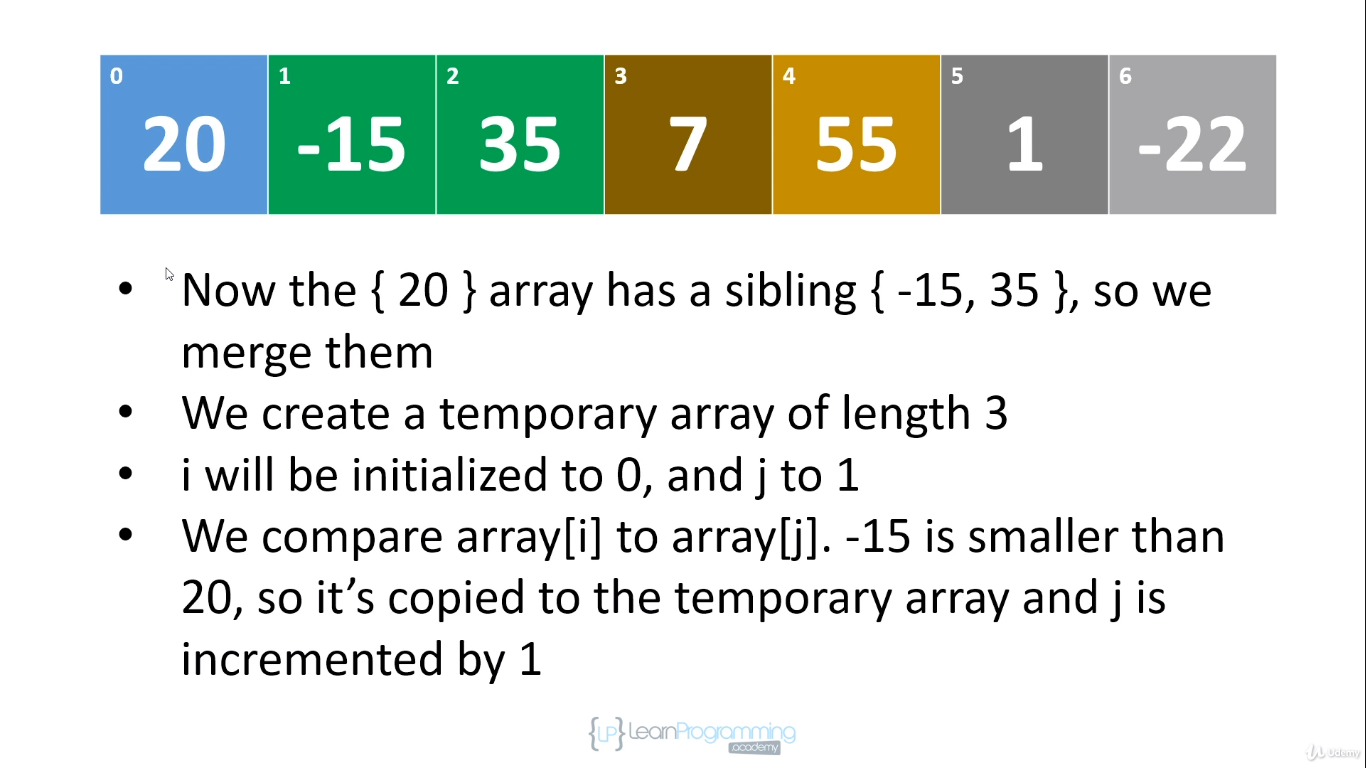


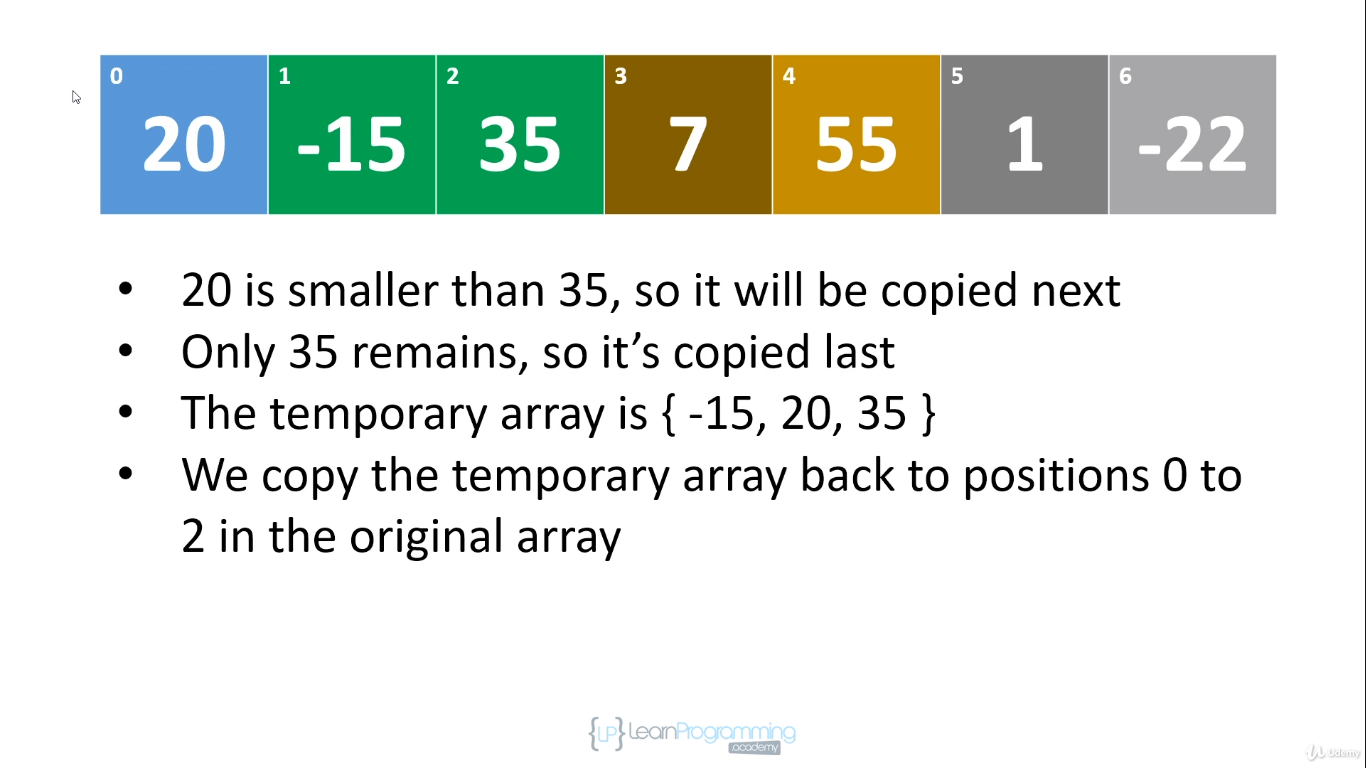


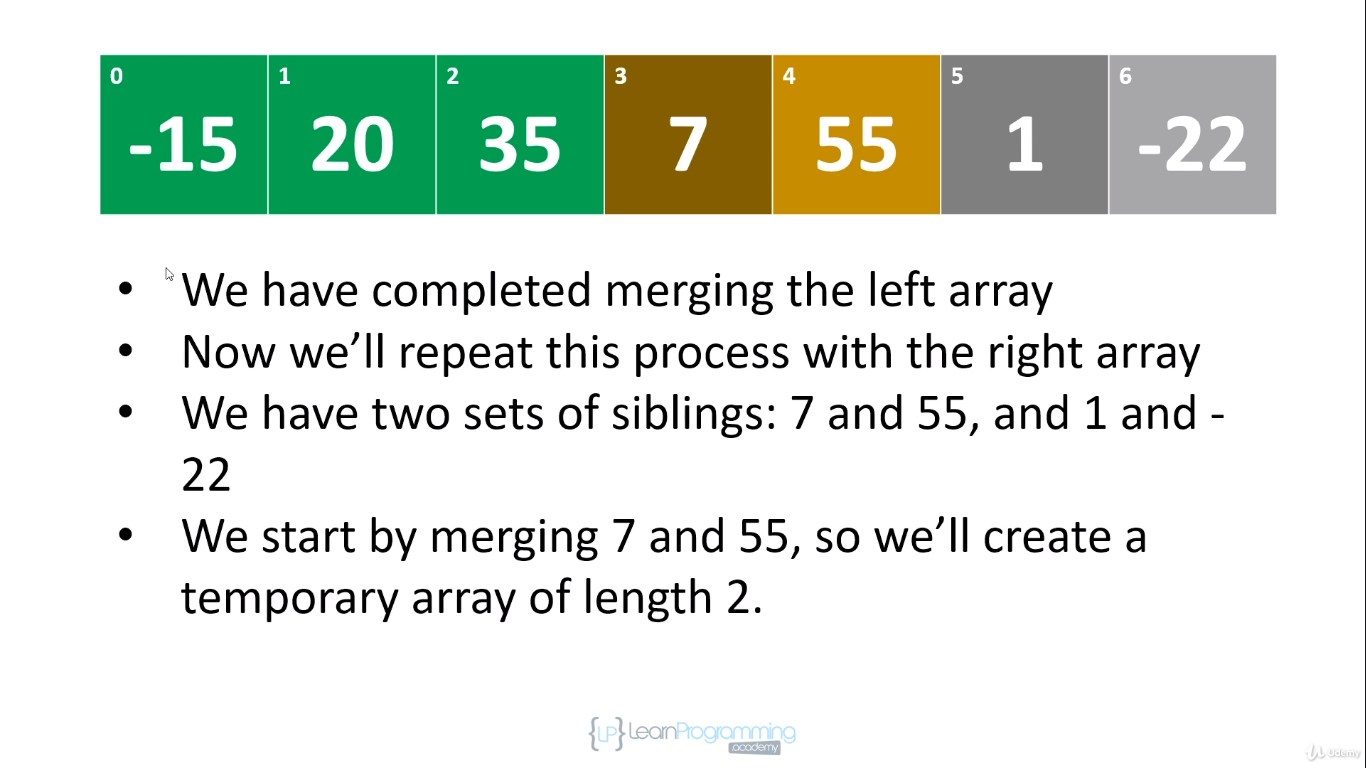


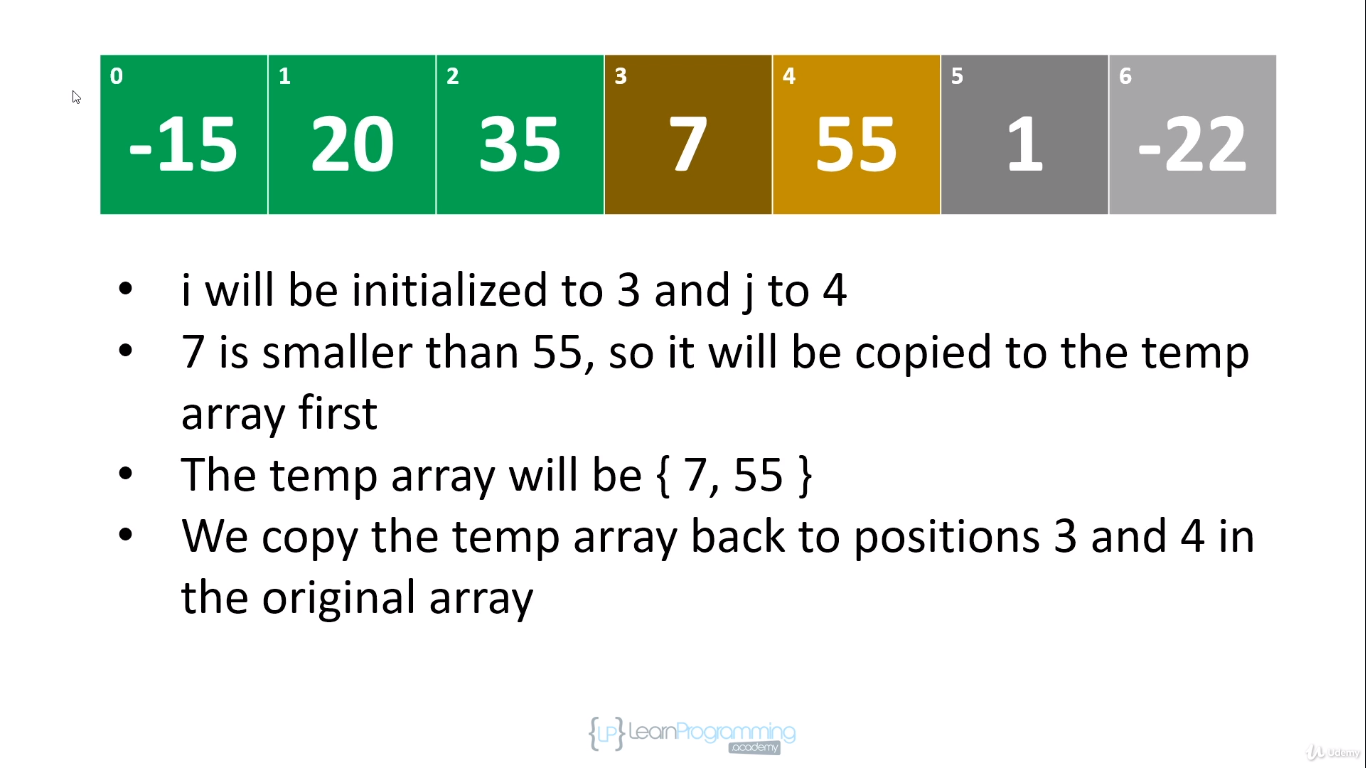


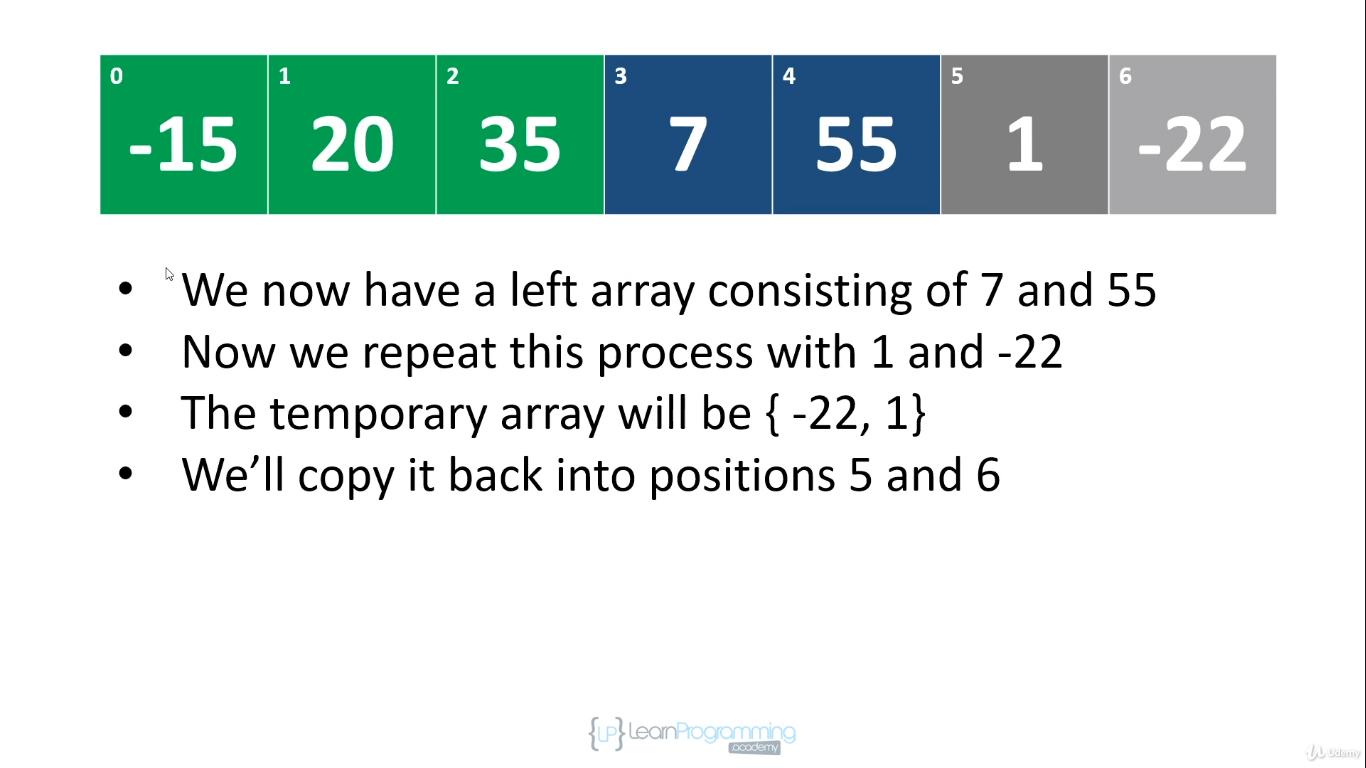


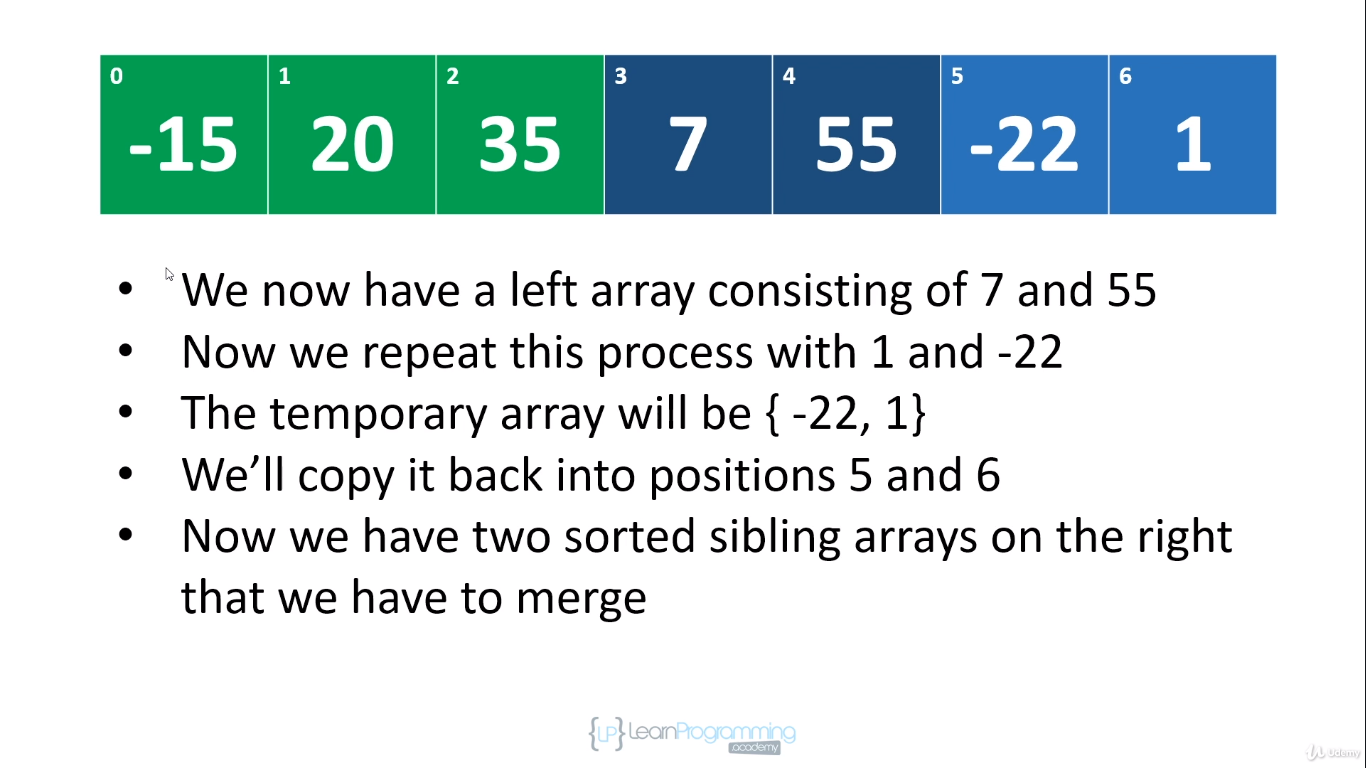


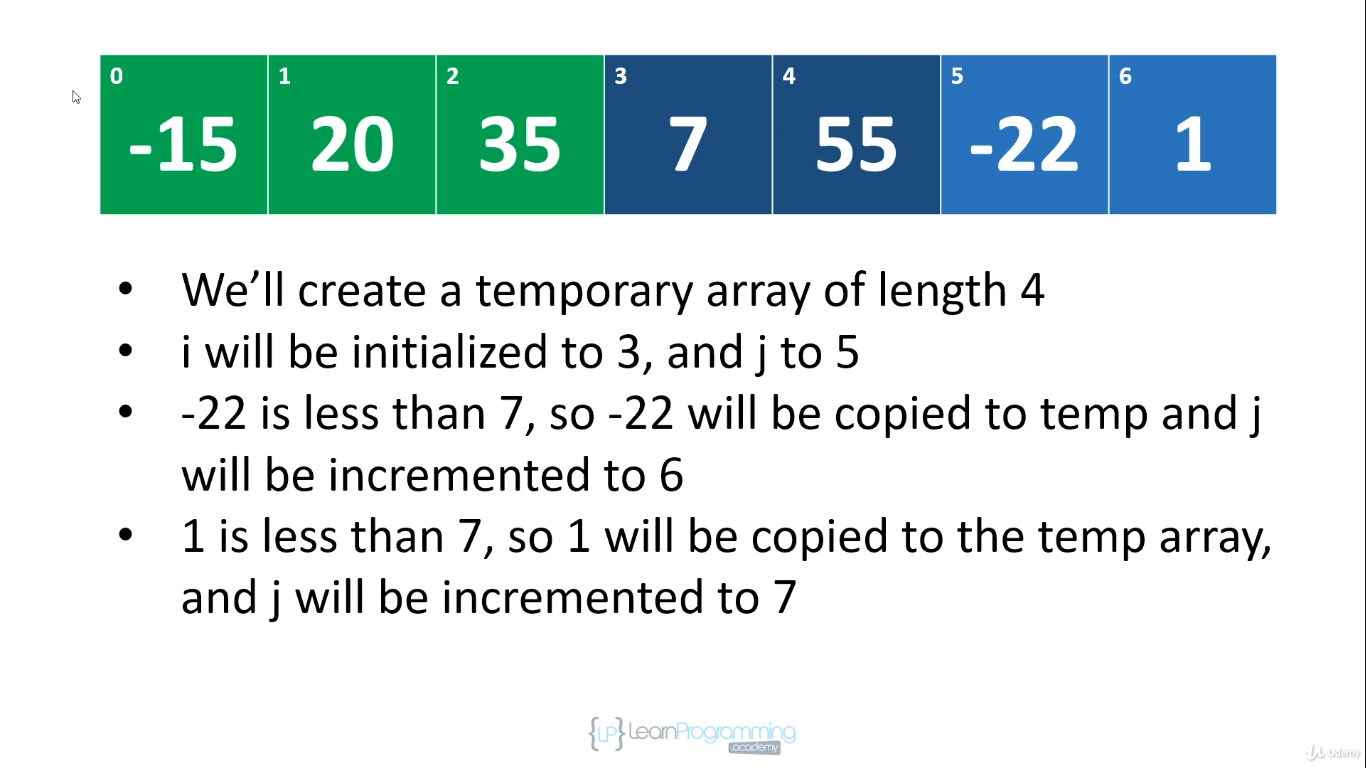




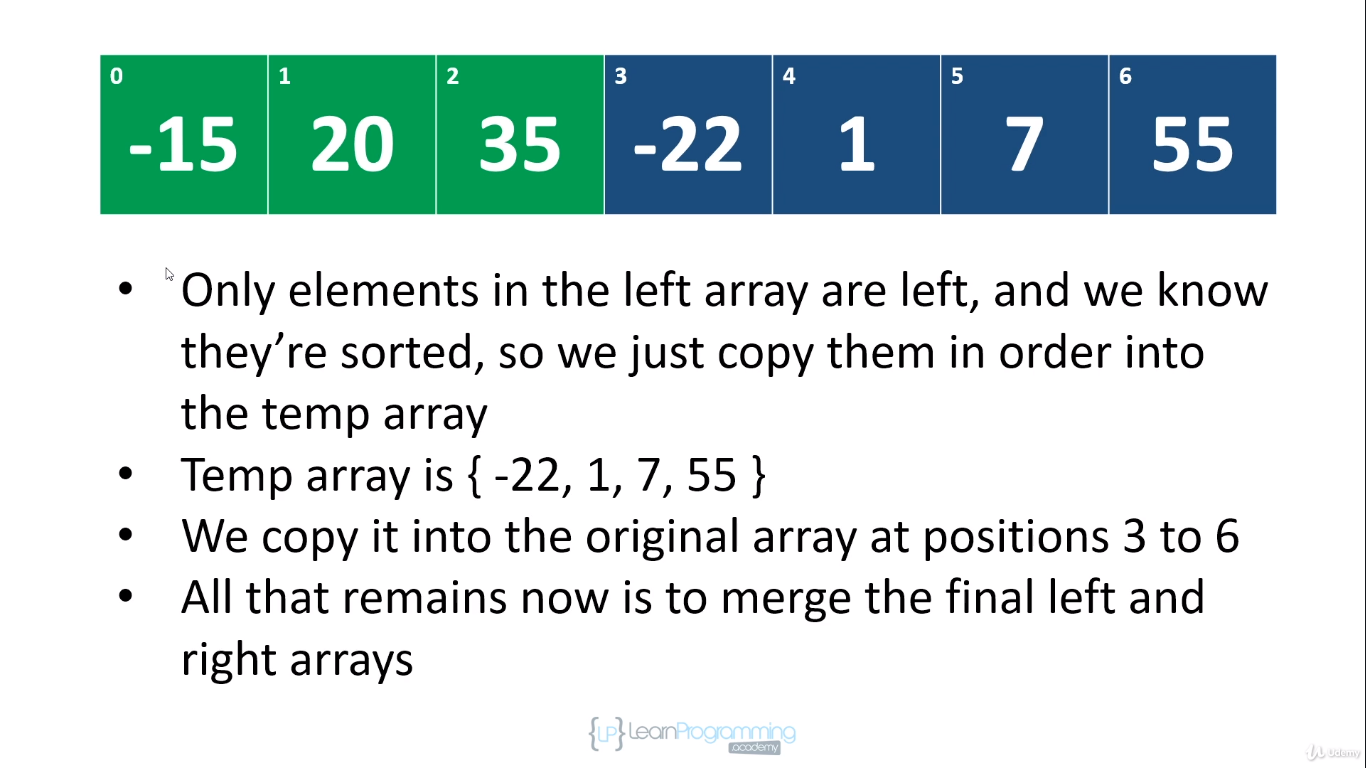


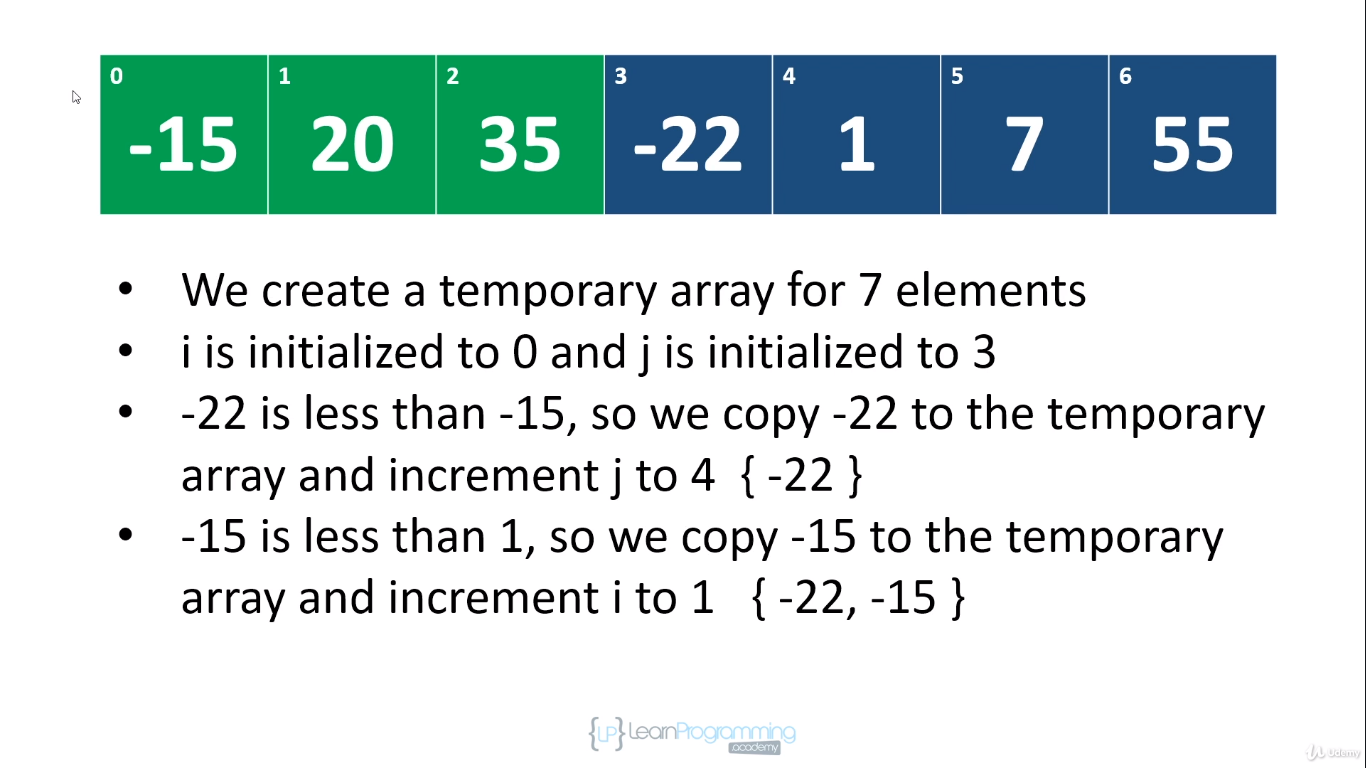


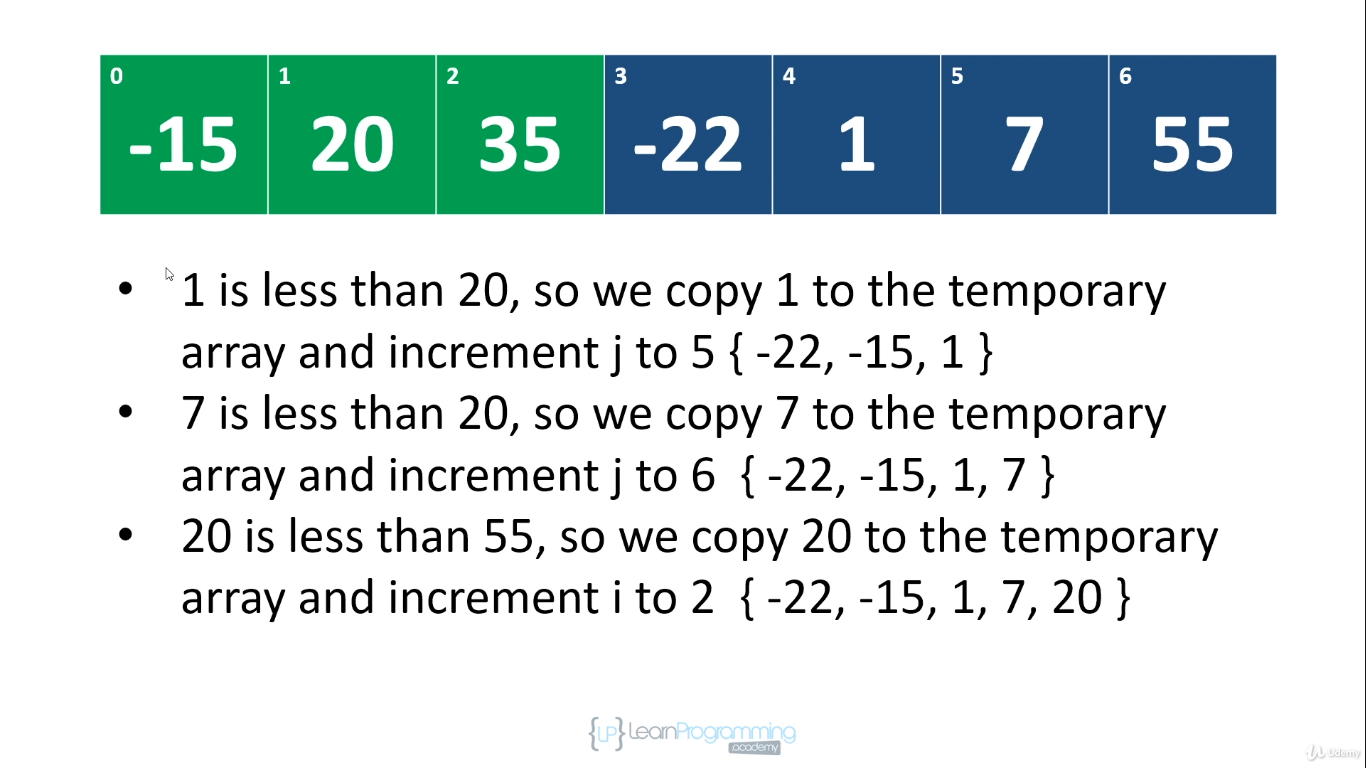


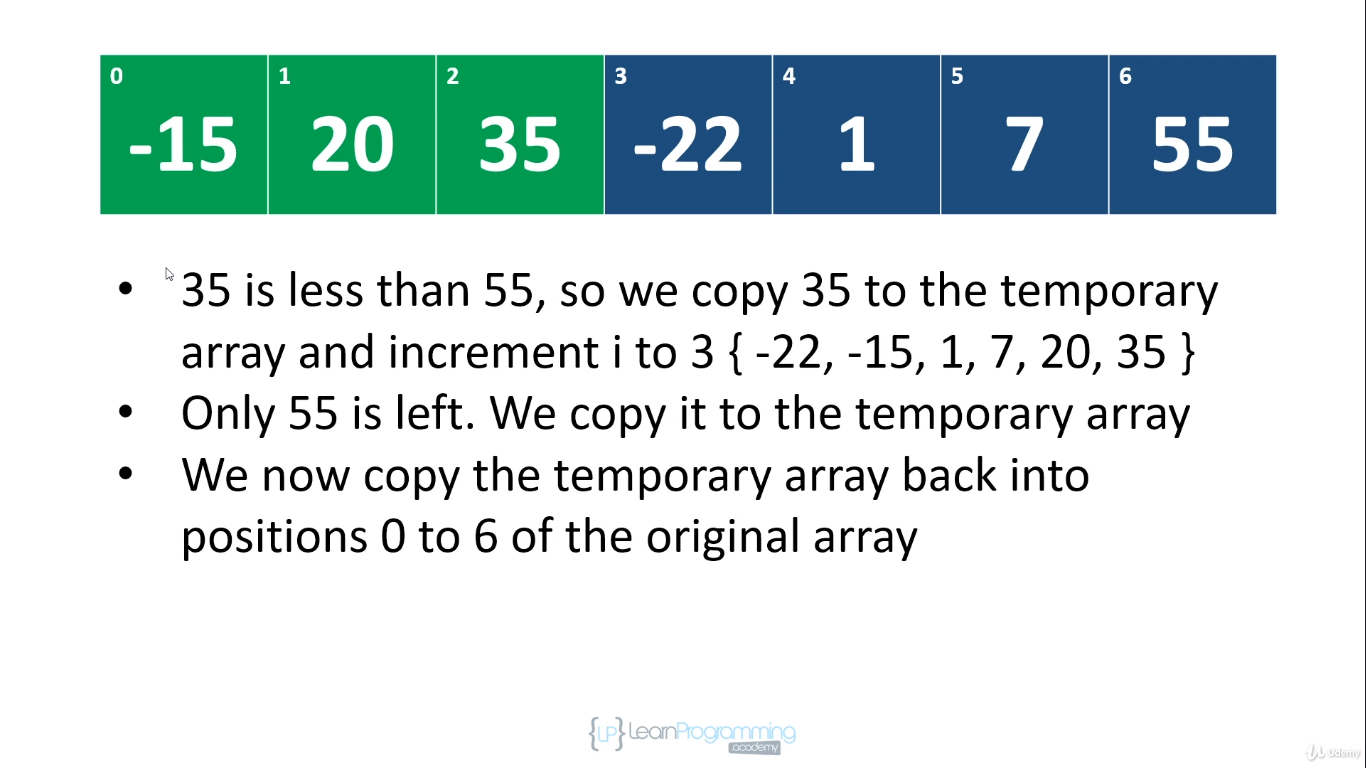






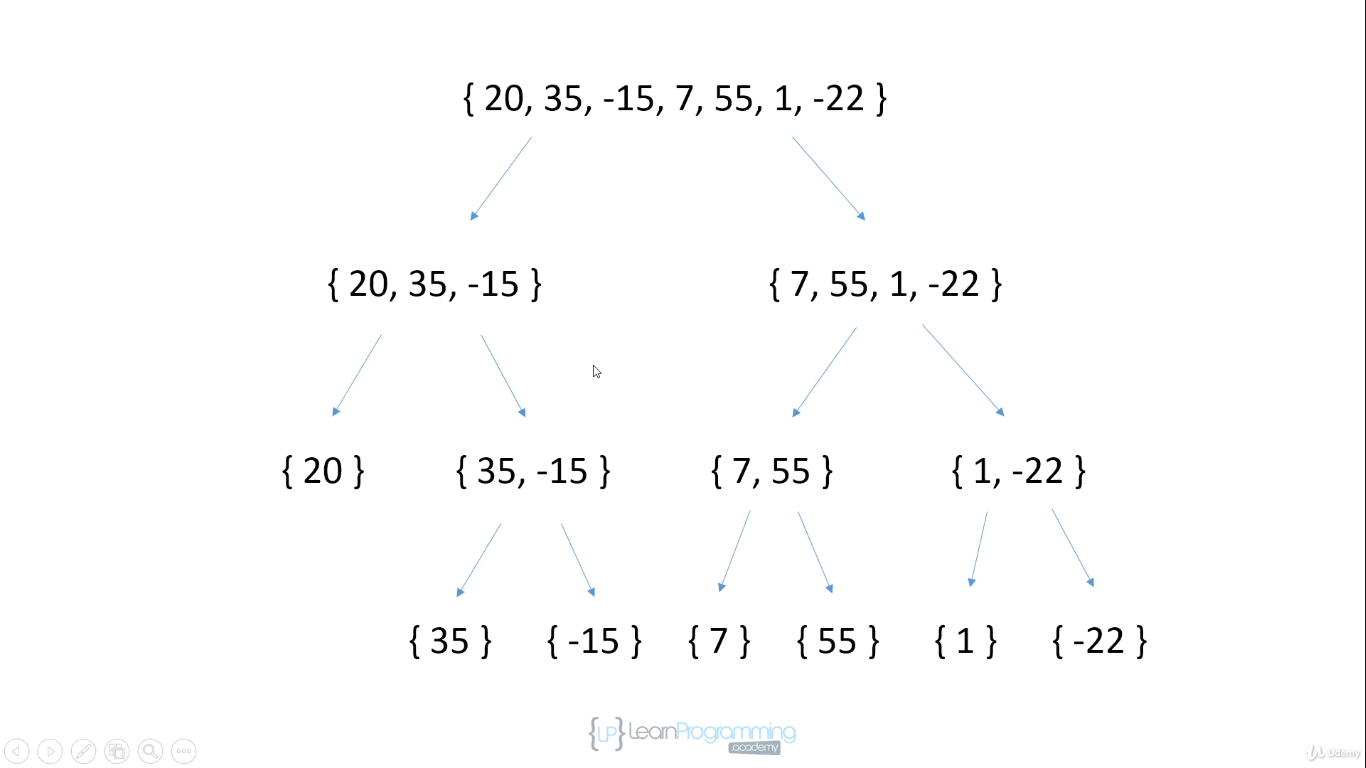








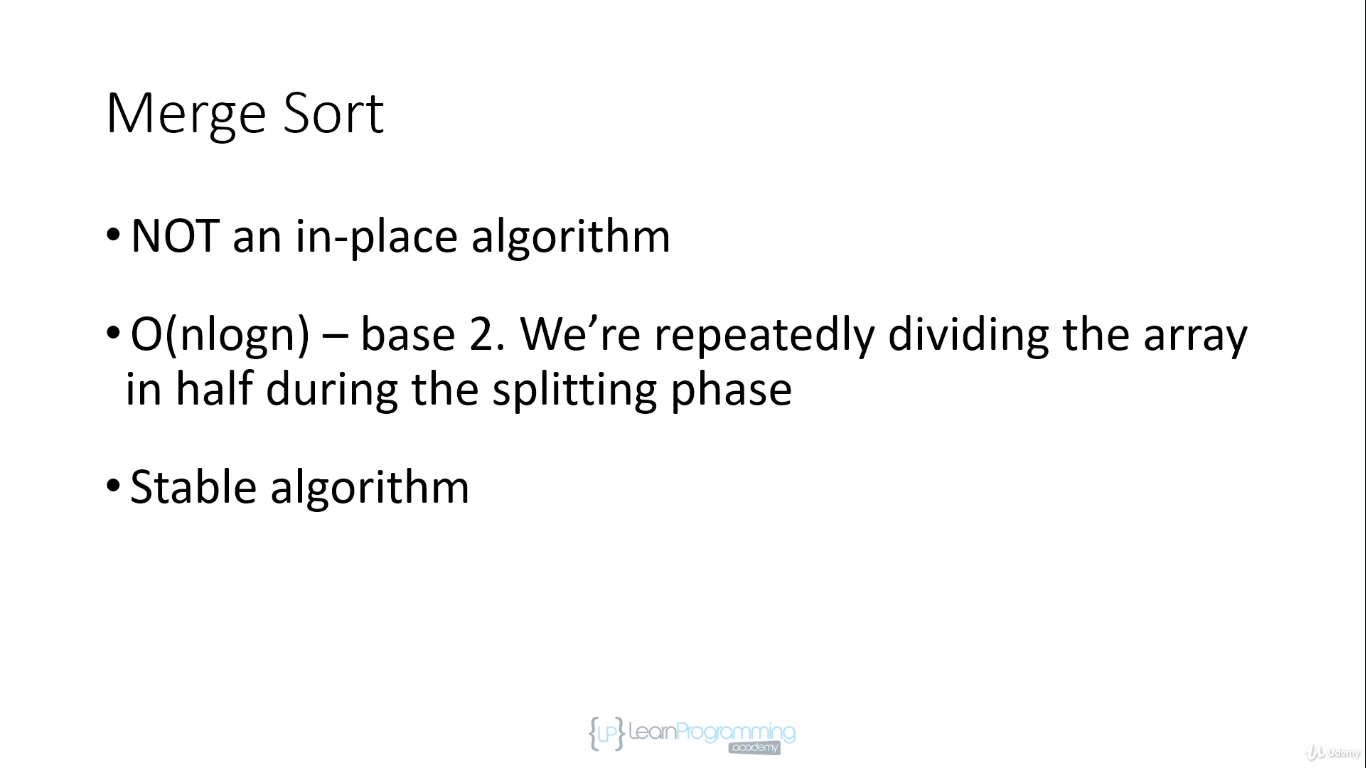
So what we had when we started we split like this



And then in merging we move from bottom to up by merging with sorting.



Take a look how algorithm performs



Splitting phase is in-place but merging is not as we use temp arrays to store sorted elements

It is stable as relative ordering of duplicate items is preserved in this algorithm.

Amount of memory these days is cheap so you can use merge sort but if memory is issue then you need to think before using it because as the no of items grows memory require to keep sorted element is temp array grows.

