

2.7.1 Two Way MergeSort - Iterative method

Merge&Merging : is a process of combining two sorted lists into a single sorted list.

A is one list and B is one list we have two sorted lists and we want to combine them also sorted way. These lists can be linked list or Array.

Merge

<u>A</u>	<u>B</u>	<u>C</u>
$i \rightarrow 2$	$5 \leftarrow j$	$\leftarrow k$
8	9	
15	12	
18	17	

How we combine them?

We take first element as I second one is J we compare them 2 is less than 5 so we take two and our I index goes to second index 8 and we add our 2 index to C list which is we defined as combined list.

Merge

<u>A</u>	<u>B</u>	<u>C</u>
2	$5 \leftarrow j$	2
$i \rightarrow 8$	9	$\leftarrow k$
15	12	
18	17	

Then we compare 8 with 5. 5 is less than 8 so C takes second element as 5.

<u>A</u>	<u>B</u>	<u>C</u>
2	5	2
$i \rightarrow 8$	$9 \leftarrow j$	5
15	12	$\leftarrow k$
18	17	

Merge

<u>A</u>	<u>B</u>	<u>C</u>
2	5	2
8	9	5
15	12	8
$i \rightarrow 18$	17	9
	$\leftarrow j$	12
		15
		17
		$18 \leftarrow k$

We called that process is merging. The property of this process is if two lists are already sorted then the result is sorted.

Time complexity: $O(m+n)$

More than two lists:

Compare first elements of all these lists and copy the one which is very smaller 2 is smaller so I can

copy 2

<u>Merge</u>			
A	B	C	D
4	3	8	2
6	5	10	4
12	9	16	18

<u>Merge</u>			
A	B	C	D
4	3	8	2
6	5	10	4
12	9	16	18

$\frac{L}{2}$

As 2 is taken then I can go to next element of D

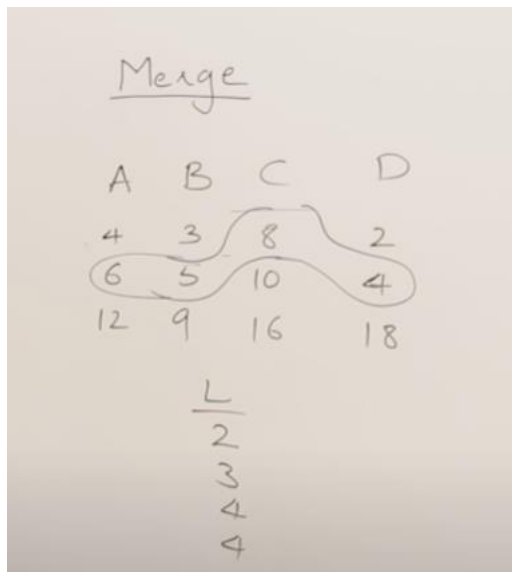
<u>Merge</u>			
A	B	C	D
4	3	8	2
6	5	10	4
12	9	16	18

$\frac{L}{2}$
3

Now smaller one is 3 let's copy 3 element of B

As 3 is taken We can pass to next

Now we compare we see smaller one is 4 and take 4 on left side.



Now 4 is smaller than we write 4 and pass to other element 18 for D.

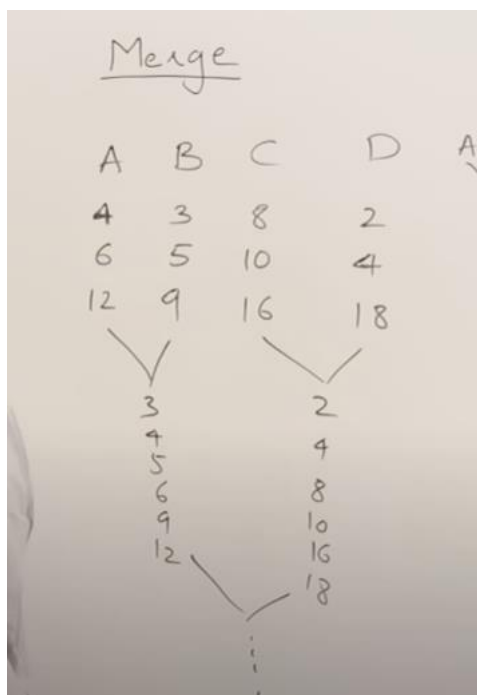
I can compare all the elements in each list and take the minimum one. We have four lists

FOUR WAY MERGING: If you merge four list at time

As many lists you have, If you follow this procedure this is called MV MERGING

TWO WAY MERGING: If you merge two list at a time.

TWO WAY MERGING:

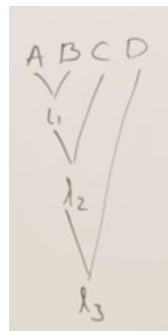


I can take two list and get one list that will be and I will take another two list and combine.

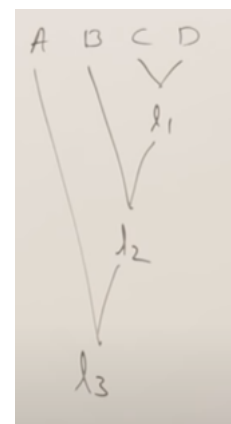
Now I can combine two list to one single list.

Other way is I take A B C D

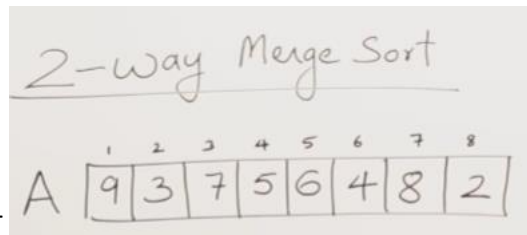
that
then



Other option is Merge like A B first then C with A B combined D to C these are other options.



If you have a multiple lists to merge you can do that a various way.

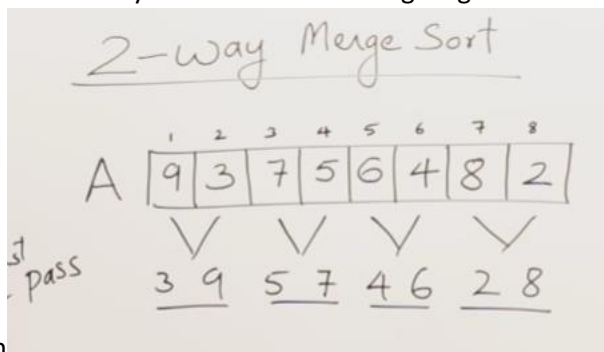


TWO WAY MERGE SORT

There are two merge sort method one is two way merge sort other one is merge sort.

Two merge sort is iterative process. Repeating procedure that is using loop and merge sort is a recursive procedures. A recursive algorithm.

When one array is given to us and we need to sort this. We assume each element in the array is list. so there are eight lists containing one element in each list. Each list has a single element. If there is a single element means that it is already sorted. I have to merge eight lists to one single lists

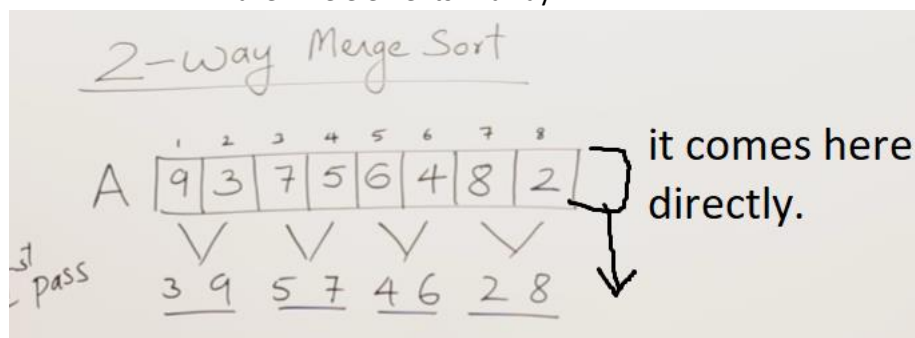


that's is how I will merge all of them

I call them lists but these are element of array. 3 is smaller than 9 so I got one list that is merging of those two is done. We compare other numbers also. Like 7 and 5 5 is smaller etc.

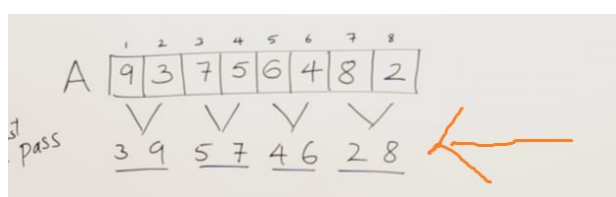
I had eight lists now I got 4 lists.

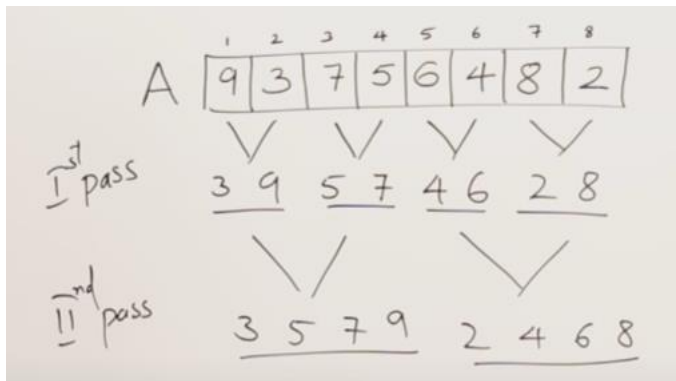
***** If I have nine elements in array



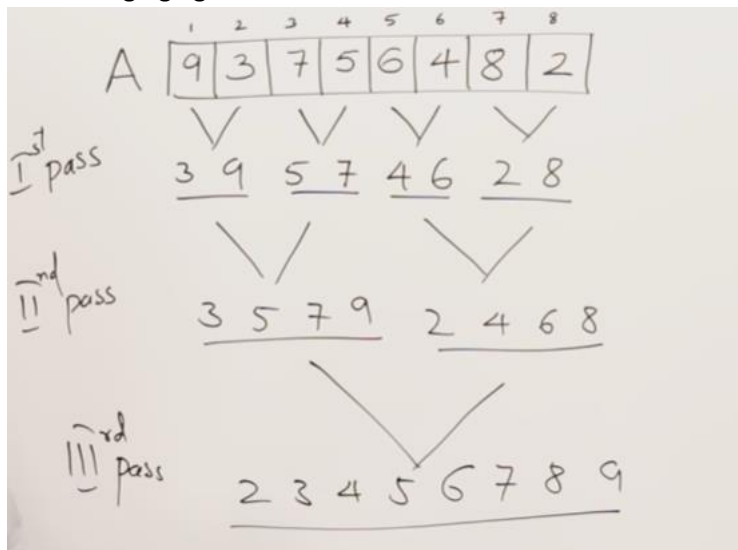
*** When I merge them where I keep elements? We need to keep our new results in new array. We need to create a new array that holds our new results. **When you merge in array, Results has to kept in different array.**

Now I have four list I have to marge them





After merging I got one list of four elements



Now we have one single sorted list by using two merge way at a time.

How much work is done in each pass?

$$\begin{array}{l}
 n \\
 n \\
 n \\
 \hline
 \text{no. of passes } \log n
 \end{array}
 \begin{array}{l}
 \frac{8}{2} = 1 \\
 \frac{8}{2^3} = 1 \\
 8 = 2^3 \\
 \log_2 8 = 3
 \end{array}$$

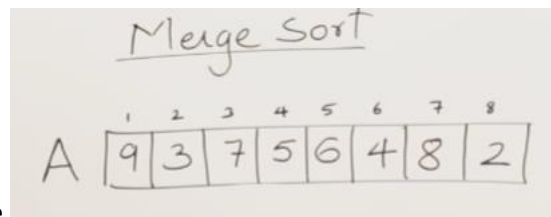
We had 8 list and divided 2 each time so We have 1.

Time complexity is $O(n \log n)$

2.7.2. Merge Sort Algorithm

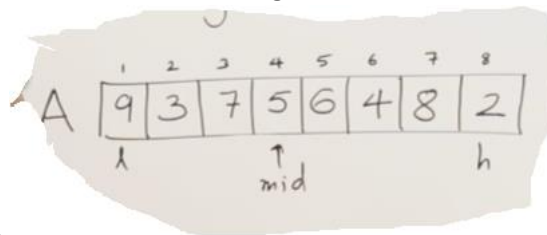
This is a recursive procedure. This is a divide and conquer procedure that's why it is a recursive algorithm.

Divide and Conquer: if a problem is large then cut problems into subproblems and problem becomes small then solve and combine the solution of the small problems or subproblems. To get the solution for main problem



Here is problem to sort this list. It's very large.

This algorithm takes two parameters beginning of the list and end of the list. We check low is less than High if low is equal to high that means we have single element. lowest less than high



means at least there are two elements.

$$\text{mid} = (l + h) / 2;$$

Then we find middle by LOW+HIGH / 2 formula what for ? this problem is large so we break it into two subproblems.

When the single array is done for sorting we are breaking into two sub arrays left is one list and right is one list so single array containing two lists so we have to merge them.

***A single array can have more than two list ? YES**

Temel olarak büyük listeyi sorted etmek zor dolayısıyla sen bunu 2'ye boluyorsun. En son item ile birinci item toplanip ikiye bolunuyor ki Ortadaki bulunsun.

When a single array is given recursively it divides that array into smaller pieces until it reaches one element.

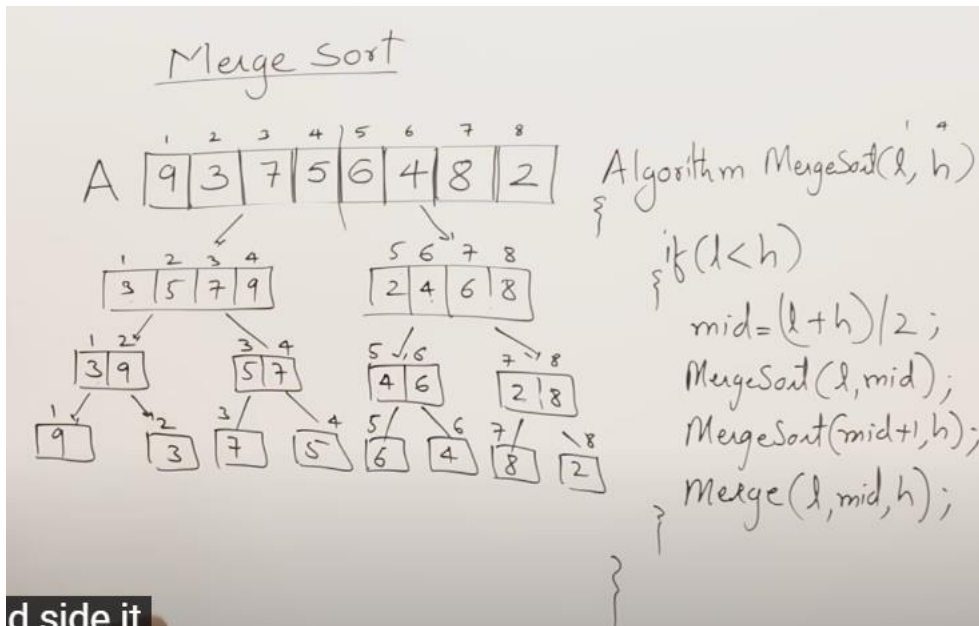
So similar to two way but in two way merge sort we just take first two element the next to the next to this is MERGE SORT recursive.

What different is only pattern is changing. The approach rest of the things are same. Results are same but this is recursive so we call that Merge sort.

Time complexity:

$$O(n \log n)$$

We have total three levels here . (n) It means for 8 element it is log



2.7.3 MergeSort in-depth Analysis

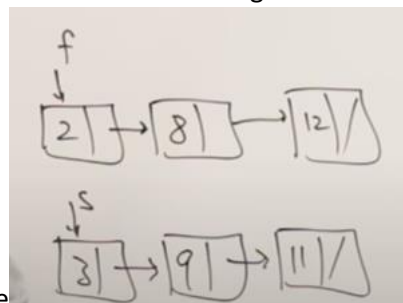
Pros

1. Large size List
2. Linked List
3. External Sorting
4. Stable

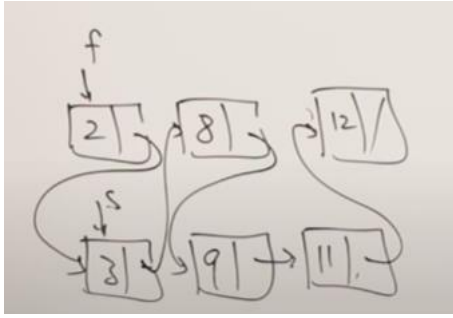
1-Merge sort is suitable for very large size list if you have millions of numbers you can sort them using merge sort .Or you have millions of records. You can sort them using merge sort.

No other sort can support very large size list.

2-Merge sort is suitable for linked lists also for example If I have two linked list merge sort uses merging. If I have two sorted linked list I can merge them without creating third linked list. Bu

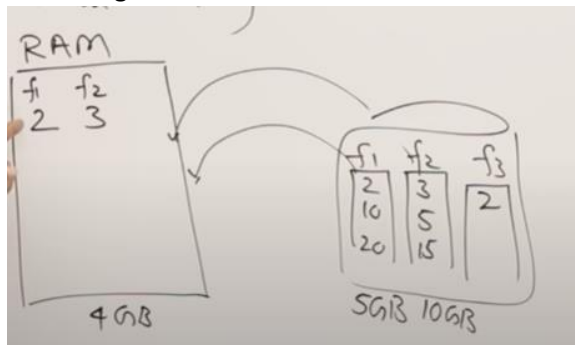


changing their links without creating third list u can merge



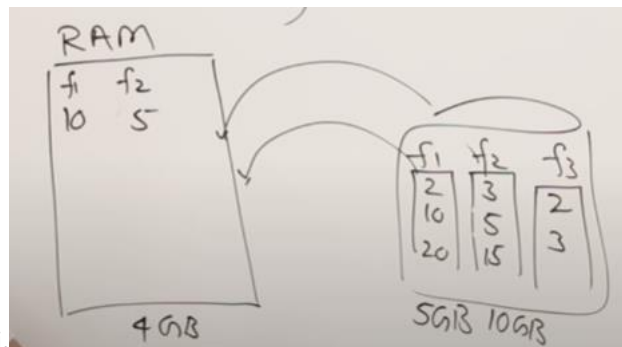
3-Merge sort supports external sorting.

External sorting : I have taken hard disk. Millions of data in hard disk. I want to merge them. If I can merge them then I can also perform merge sort because merge sort is also merging. All of files should come to ram but ram is 4 GB so I cannot take entire files to memory. I can bring part of a file and merge and I can store that in F3. Parcalara bolerek getirebilirim. First we take 2-3 and compare 2 is small then we write that



on F3 and so on.

Simdi 2 silip yerine 10 yazacagiz



Cunku 2 ile isimiz bitti.

4-MERGE SORT IS STABLE

If I have a list which is duplicates. 8 is duplicated it is appearing two times.

So if you see here after sorting duplicated values are maintained.

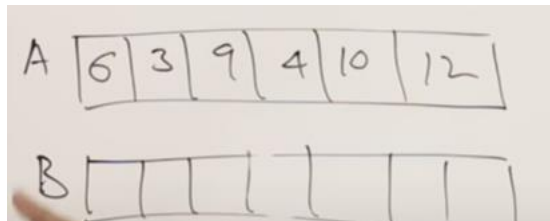
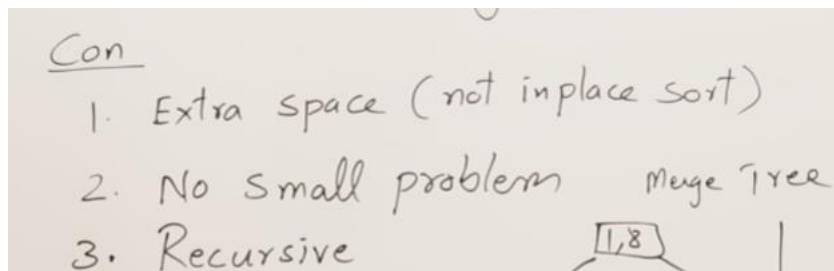
Ravi	Sunil
8	6
4	3
3	8
4	5
5	9
6	8
8	8
8	9

Imagine this is a record of students marks.

We can keep Sunil and ravi's marks are together in our data record because merge sort

Does not delete same Two results.

***If they are some data arrangement and you want to sort it then if there duplicates we can lost our duplicated values BUT MERGE SORT DOES KEEP OUR DUPLICATED VALUES



I want to sort that. sorting algorithm needs to rearrange the elements within the same array only. And sort them. If it is rearranging in same array we call it in place sorting.

In place sorting: Rearranging in same array

Merge sort is not in place sorting. The result must kept in separate array. Merge results cannot not be in same array. That means we need EXTRA SAME SPACE FOR THIS PROCESS. ONLY IN CASE OF ARRAY NOT FOR LINKED LISTS. So merge sort is not suitable for arrays it means extra space. So in place sorting is not possible.

2-NO SMALL PROBLEM

For merge sort problems are large. We break large problems to small subproblems. And when it small there is single element we don't do anything there so it's not having small problem. Because it is divide and conquer FOR SMALL SIZE LISTS MERGE SORT IS SLOWER COMPARED TO OTHER SORTS.

We can say that Merge sort is faster than Insertion sort. But IF LIST SIZE IS VERY SMALL MERGE SORT WASTES A LOT TIME IN RECURSION. AND IT IS BECOMING SLOWER TO INSERTION SORT. When you run a two algorithm machine in computer. we find that merge sort is slower if lists size is small. So IF we have less than 15 elements in list; Merge sort would be slower. After that If lists is increases, then merge sort would be fine.

So; If list is small we assign it to insertion sort. So we use insertion sort as a part of merge sort.

So merge sort will not sort of small lists. It takes helps from insertion sort.

But why insertion sort?

Because insertion sort is also stable . there is the reason we use insertion sort. Along with merge sort.

Is there any algorithm is stable ? YES

Bubble sort. Bubble also is like insertion sort. so we can use Bubble sort instead of Insertion sort.

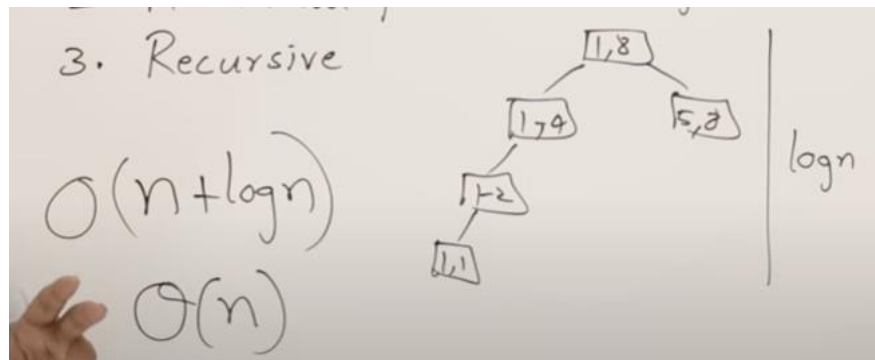
But Merge sort and Insertion sort are suitable for linked lists.

3-RECURSIVE

All recursive algorithms use stack. They stack in the memory. Merge sort also uses a stack.

The maximum size of stack depends on the tracing height of our tracing tree Merge sort.

$\log n$ is stack space



N is extra space.