

Merge sort

- merge sort is a recursive algorithm that works by cutting a list in half repeatedly, sorting the halves and merging the sorted halves back together into one list

Merge sort (2)

83	12	42	68	70	33
0	1	2	3	4	5

- start by cutting the list in half

83	12	42	68	70	33
0	1	2	3	4	5

- cut the halves in half
 - continue cutting until the lists are size 2 or smaller

83	12	42	68	70	33
0	1	2	3	4	5

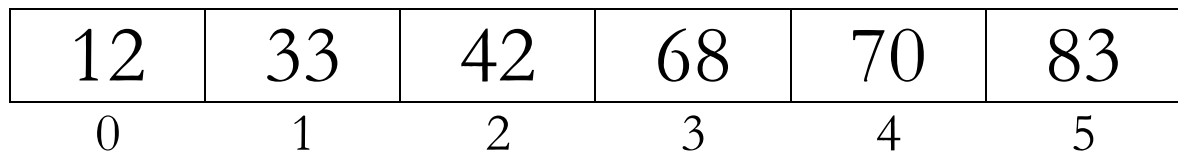
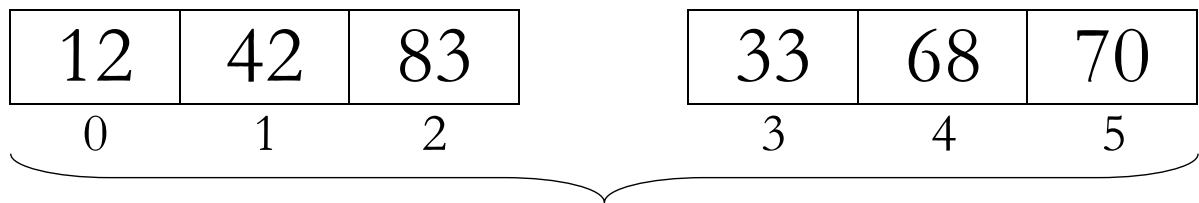
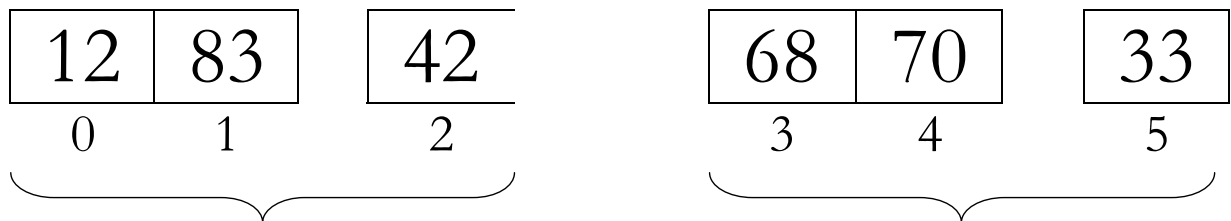
- sort the lists:

83	12	42	68	70	33
0	1	2	3	4	5



Merge sort (3)

- merge the lists back together:



- and we're done!
- but ... how do you merge two sorted lists?

Merging

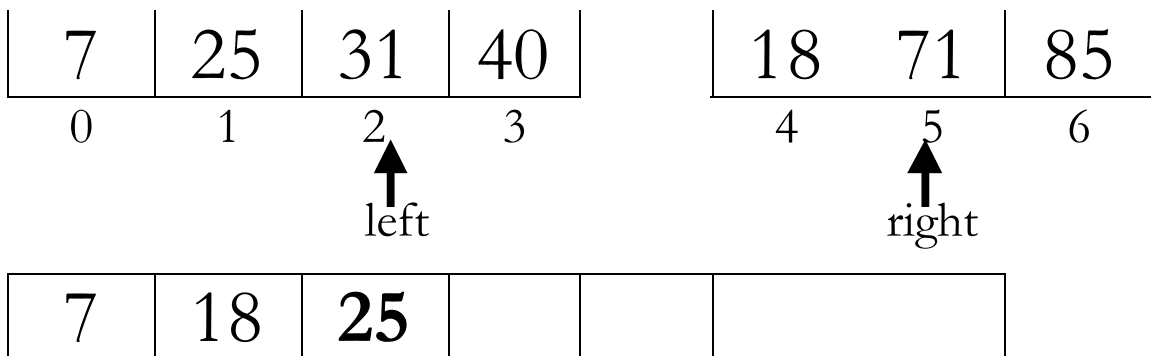
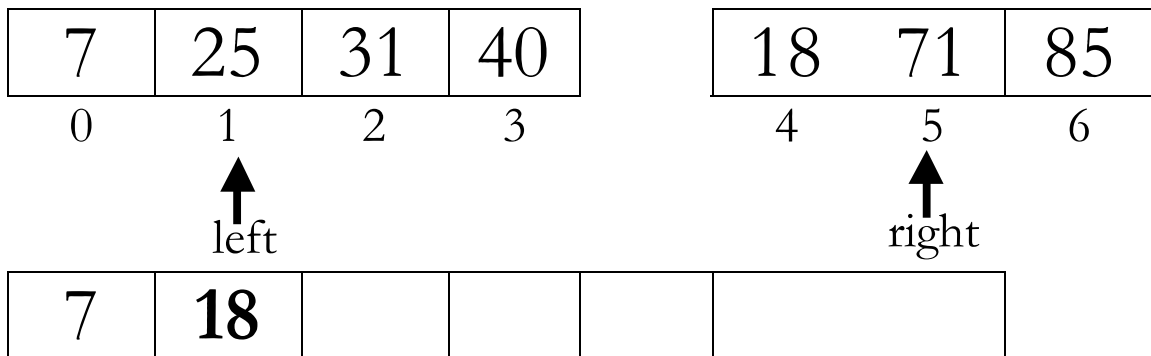
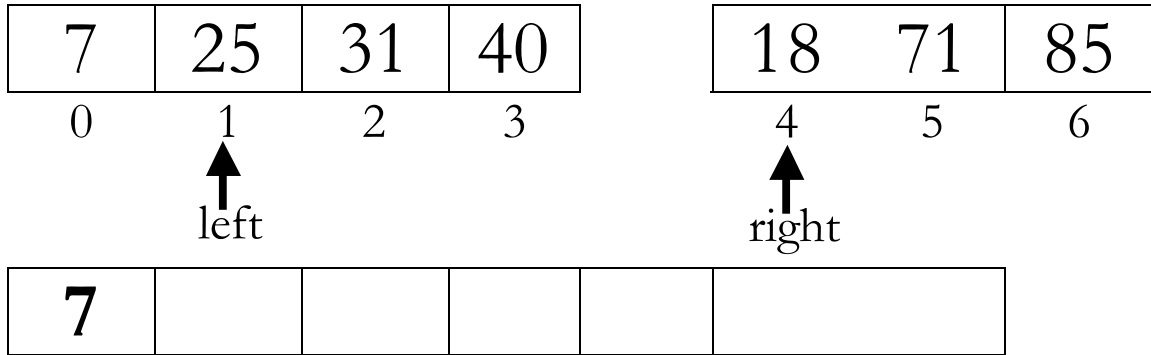
- merging the sorted lists is the hardest part about merge sort

7	25	31	40	18	71	85
0	1	2	3	4	5	6

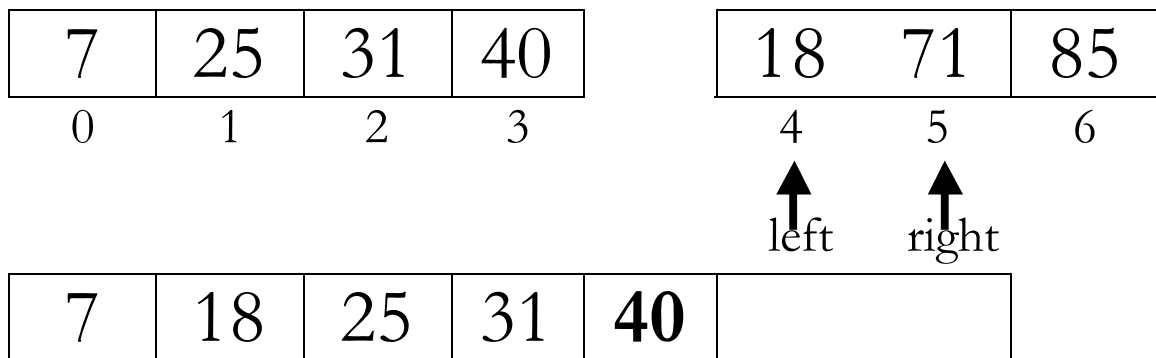
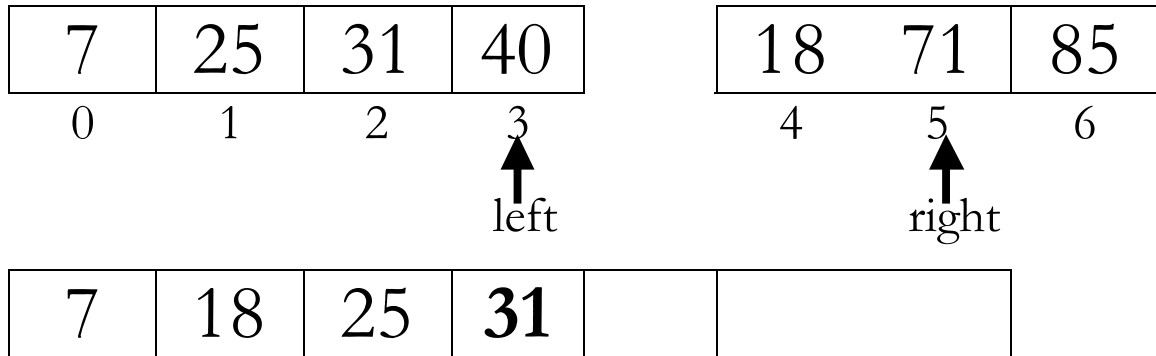
- compare the elements at the front of the left and right lists, and put the smaller element in a separate list
 - then look at the next elements

7	25	31	40	18	71	85
0	1	2	3	4	5	6
↑				↑		
left				right		

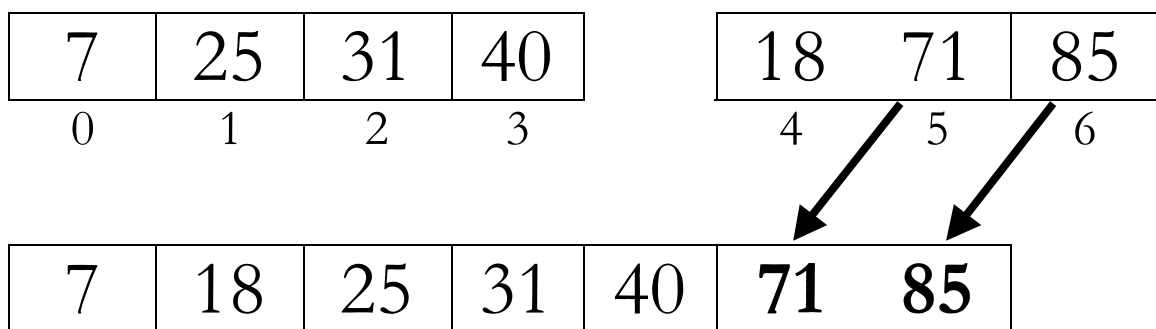
Merging (2)



Merging (3)

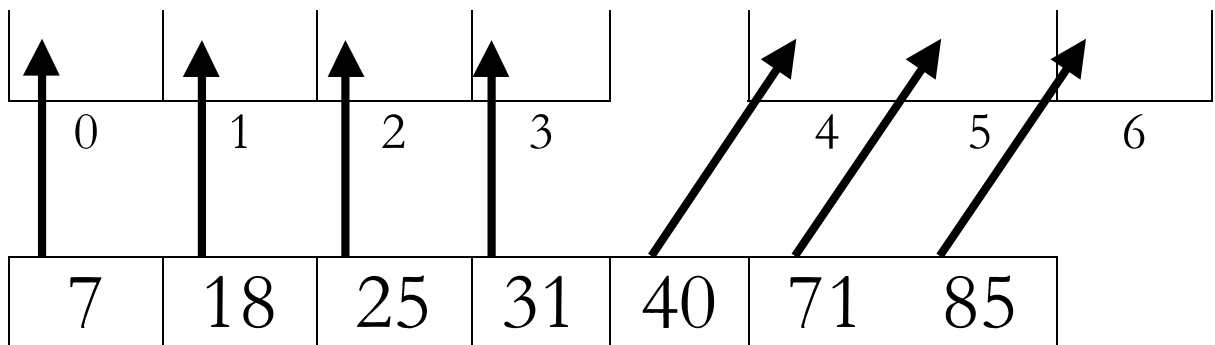


- the left list is now done – we can just copy over the rest of the right list



Merging (4)

- one more thing: we need to copy the sorted list back into the original list
 - then discard the temporary list



7	18	25	31	40	71	85
0	1	2	3	4	5	6

- and we're done!
- now ... how do you code this?

Merge sort code

- the basic steps:
 - cut your list in half
 - sort each half (with merge sort)
 - merge them back together
- merge sort is a recursive function
 - inputs: an **array** and two indexes to indicate the **start** and **end** of the array (like binary search)
 - base case: (the array “size” (indicated by start and end) is less than or equal to 2): sort the array
 - recursive case: cut the array into two pieces, call merge sort again, then merge the two pieces

Now, the code?