Merge Sort

Is going to do better than selection sort and bubble sort, that is better than n^2.

Sort left half of numbers
Sort right half of numbers
Merge sorted halves

" If I want to sort all of these things, go sort the left half, then go sort the right half, and then merge the 2 together".

Each of these lines is sorting a smaller piece of the problem.

Eventually, we'll be able to co. pare this down into something that doesn't go on forever because, in fact, in merge sort, there's a base case too.

The base case in any use of recursion to make sure that you don't mindlessly call yourself forever. ← If only one number

Quit → done

Else

Sort...

→ You've got to stop at some point.

What does it mean to merge two lists,

two halves of a list,

just because this is apparently going to be a key ingredient

2 4 5 7   0 1 3 6

The left half at the moment is already sorted

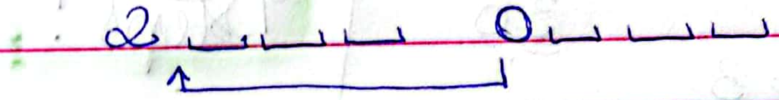
The right half is also sorted as well.

That's a good thing, because it means that theoretically, I've sorted the left half already and the right half before we began.

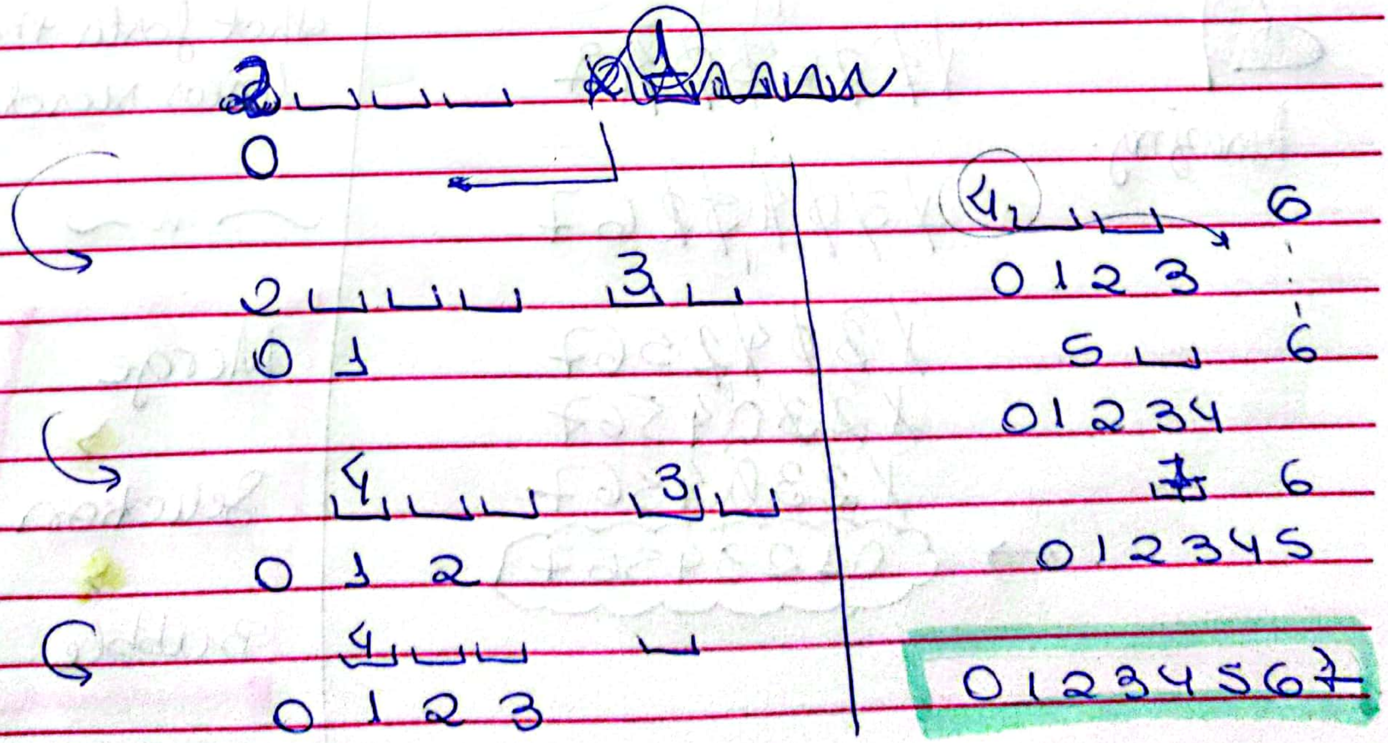
+ I just need to merge these 2 halves

I'm going to turn over MOST of the numbers, except for the 1º numbers in each of these halves.

here's 2 halves here, left and right:



→ How do I merge these 2 lists together?
which one should presumably come first?
~ the smaller one ~



5 2 7 4 1 6 3 0

5 2 7 4 6 3 0
1 5

2 7 4 6 3 0
1 5 2

7 4 6 3 0
1 5 2 6

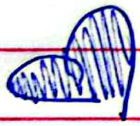
7 4 3 0
1 5 2 6 3

7 4 0
1 5 2 6 3

7 4
1 5 2 6 3 0 4 7

1 3 0 4 5 2 6 7

1 3 0 4 2 5 6 7
1 2 3 0 4 5 6 7
1 2 3 0 4 5 6 7
→ 0 1 2 3 4 5 6 7



Amazing.

Algorithm
5 2 7 4 1 6 3 0

2 5 | 4 7 | 1 6 | 0 3

2 4 5 7 | 0 1 3 6

0 1 2 3 4 5 6 7

4 steps!

(log n) < (n)
↓
that faster than linear search (n)

~ n ~

Merge
>
Selection
>
Bubble