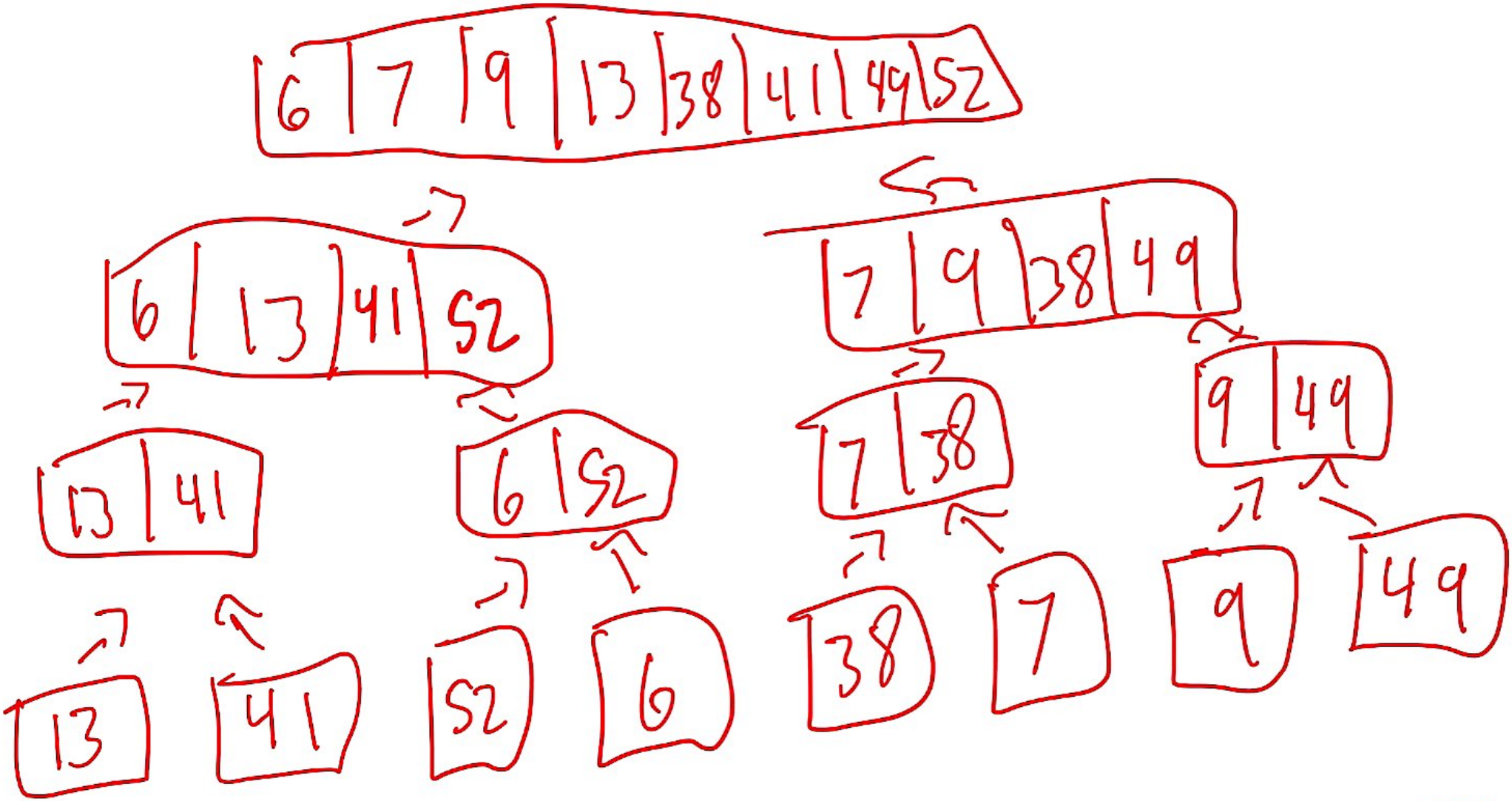
Question 2 merge sort

A = {13, 41, 52, 6, 38, 7, 9, 49}



Question 3 Bubblesort

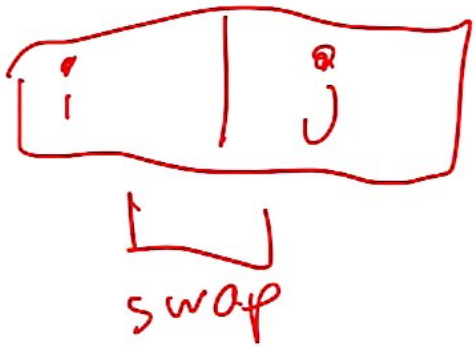
```
for i = 1 to A.length - 1
    for j = i + 1 to A.length
        if A[j] < A[i]
            exchange A[j] with A[i]
```

a)

initialization: will only iterate if array.length ≥ 2 doesnt run otherwise since one element doesnt need to be sorted

maintainance: in first iteration the first element becomes sorted

- Only swaps neighbors until the end

for example 

where A[i] > A[j]

we swap until we reach the end of the array, and keep incrementing i with current index to be sorted

Termination: - we exit once we reach the end of the first loop which is when i = A.length - 1

- since we finished the outter loop, which means we iterated through all indeces of the array as well as compared them with all values to its right so hence we can assume its sorted from small to big

b) Best and worse case is $\Theta(n^2)$ since we iterate through all values twice, while insertion sort Best case is $\Theta(n)$ and worse is $\Theta(n^2)$ the same as insertion sort. Overall insertion sort is more efficient

Here despite being sorted $T(n)$ is $\Theta(n^2)$ with Bubble sort

Question 4 (2 credit)

Draw a recursion tree for the recurrence $T(n) = 2T(n/4) + cn^2$ for level 0,1 and 2. Show the per-level cost of the tree.

$$T(n) = \begin{cases} c, & n = 1 \\ 2T(n/4) + cn^2, & n > 1 \end{cases}$$