Tim Marder
APCS2 Pd02
HW#06 -- How Fast Are Your Turtles?
2018-02-13

Win

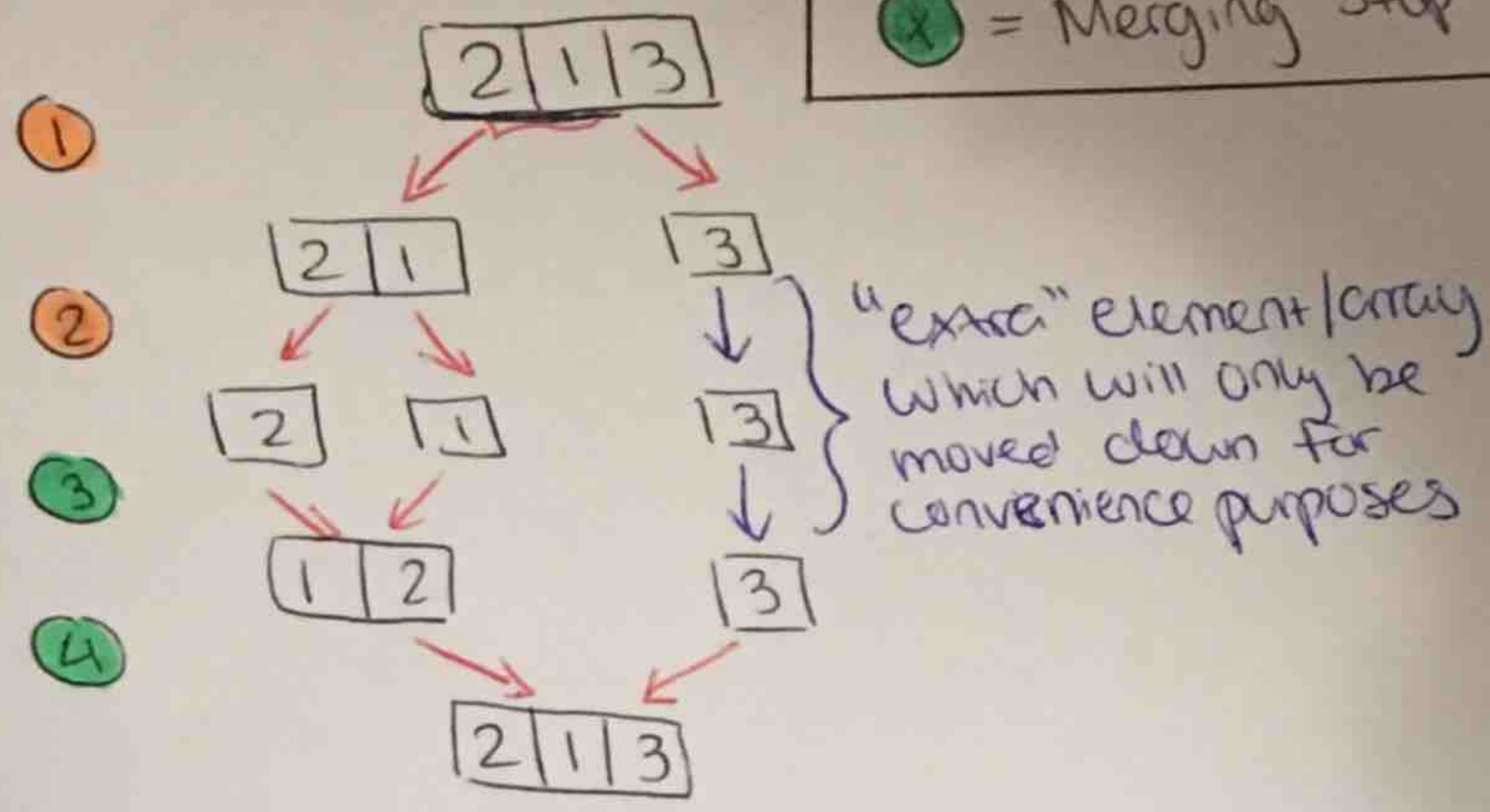
Key

→ = Split or Merge

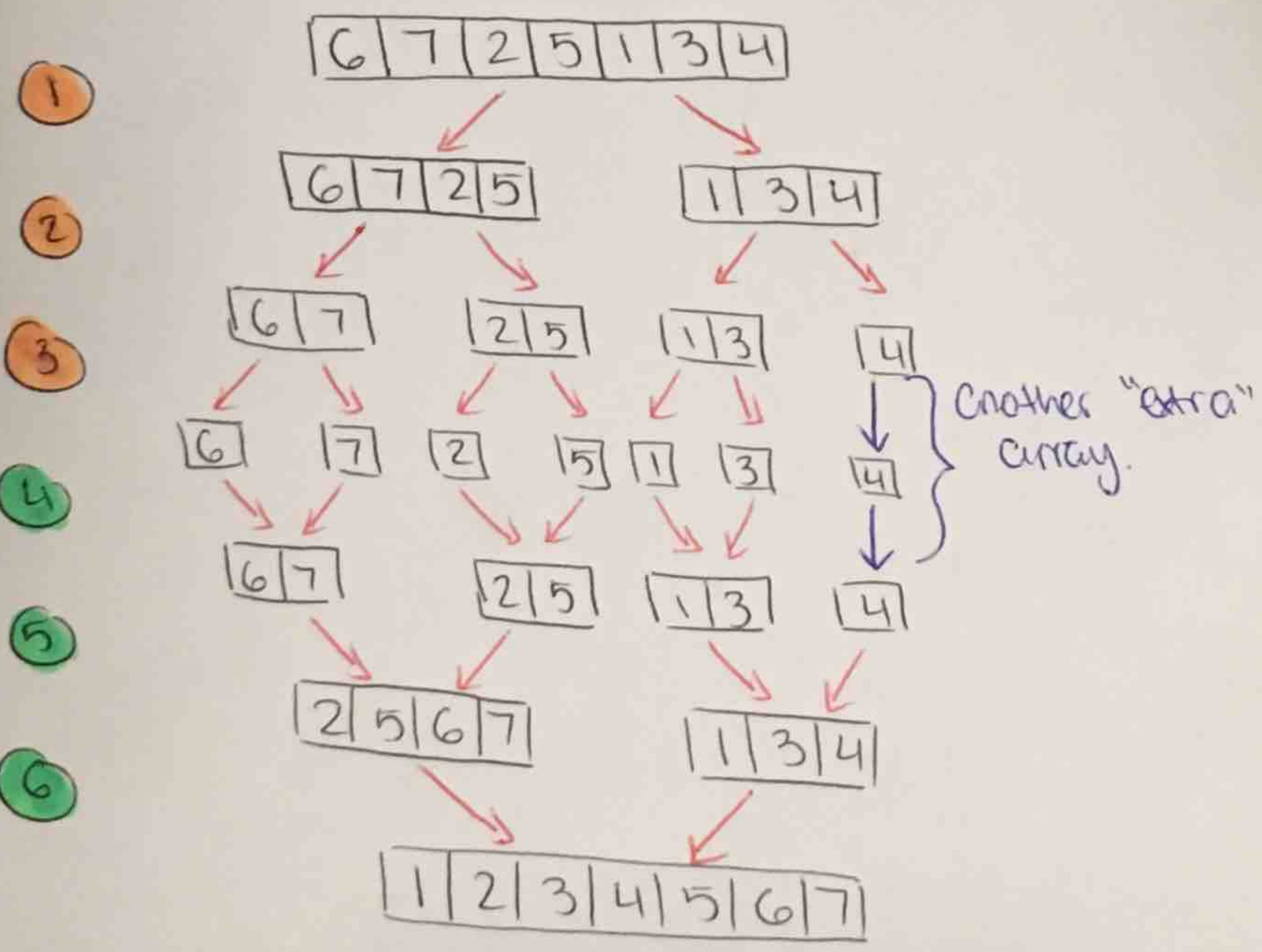
→ = No action; just brought down elements (arrays)

(1) = Splitting Step

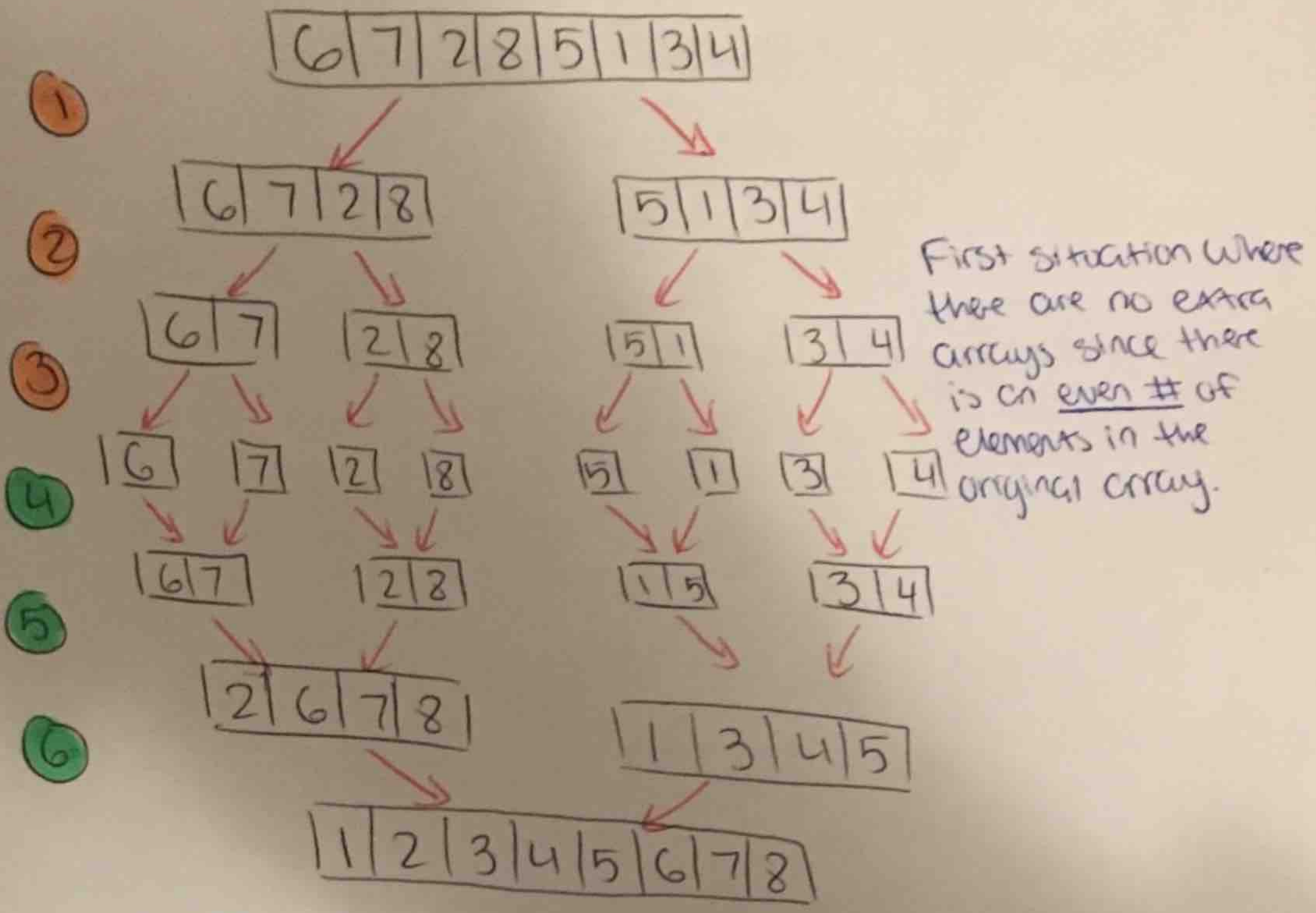
(4) = Merging Step



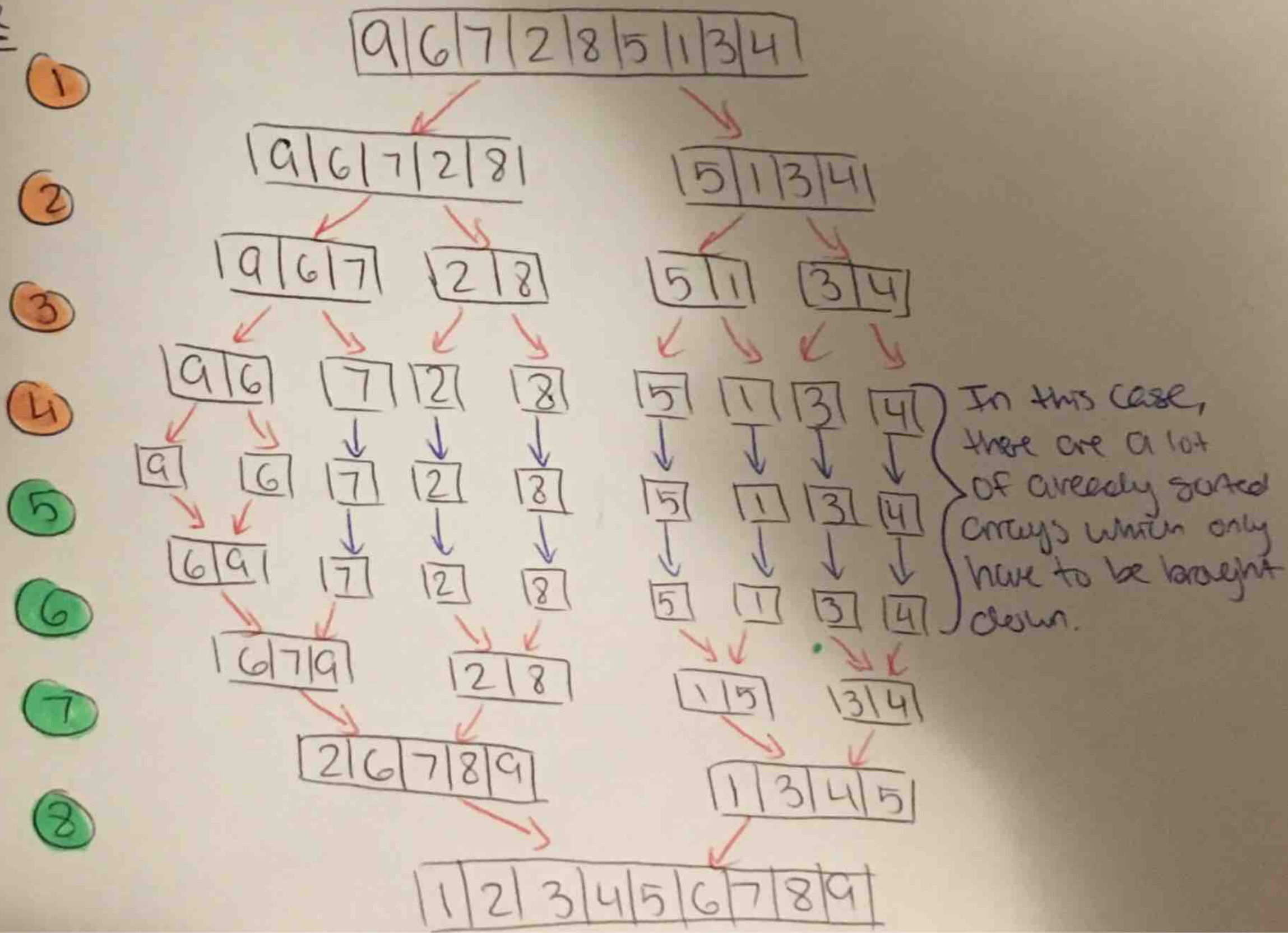
Tim :)



Jeremy

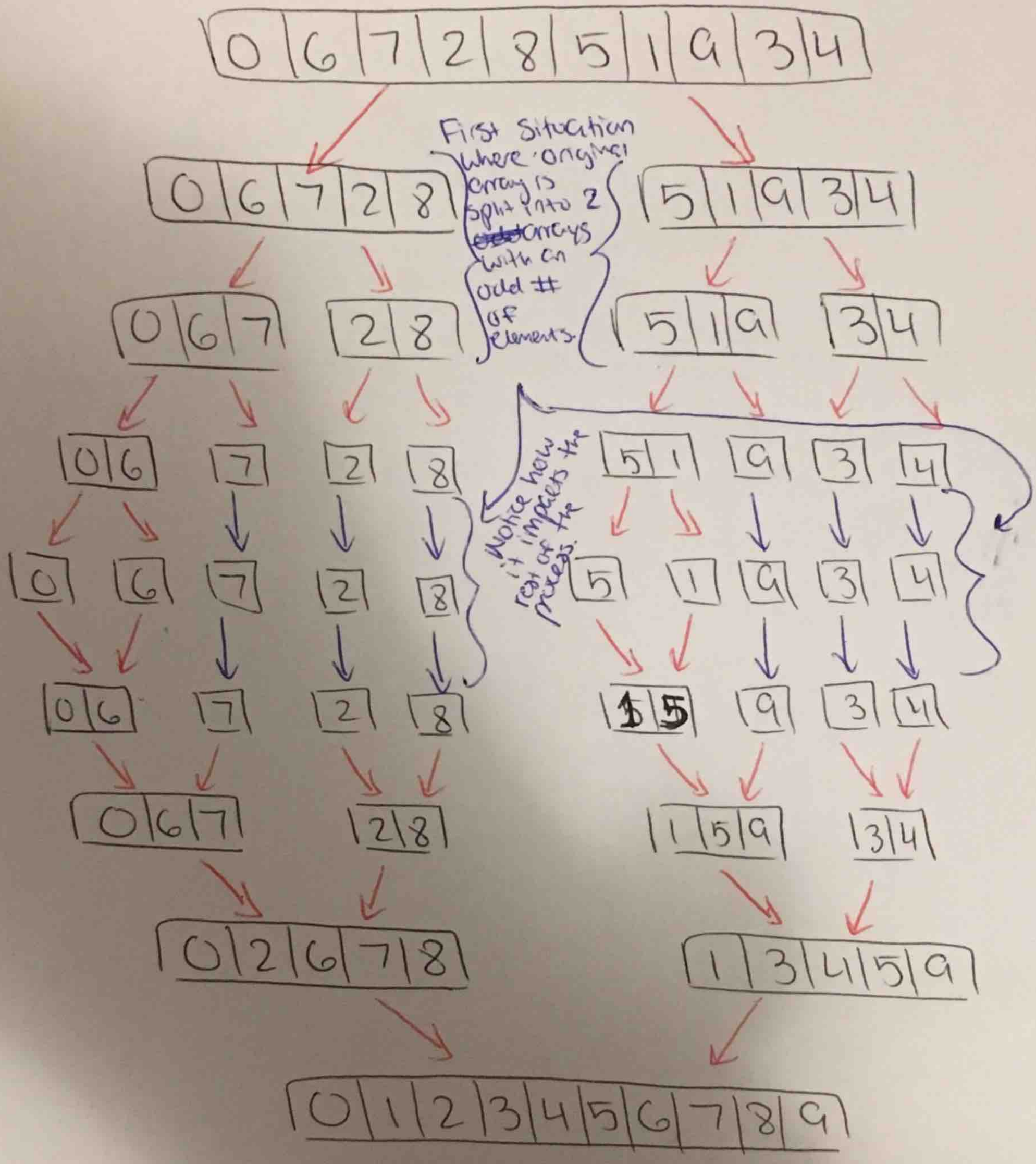


Régine

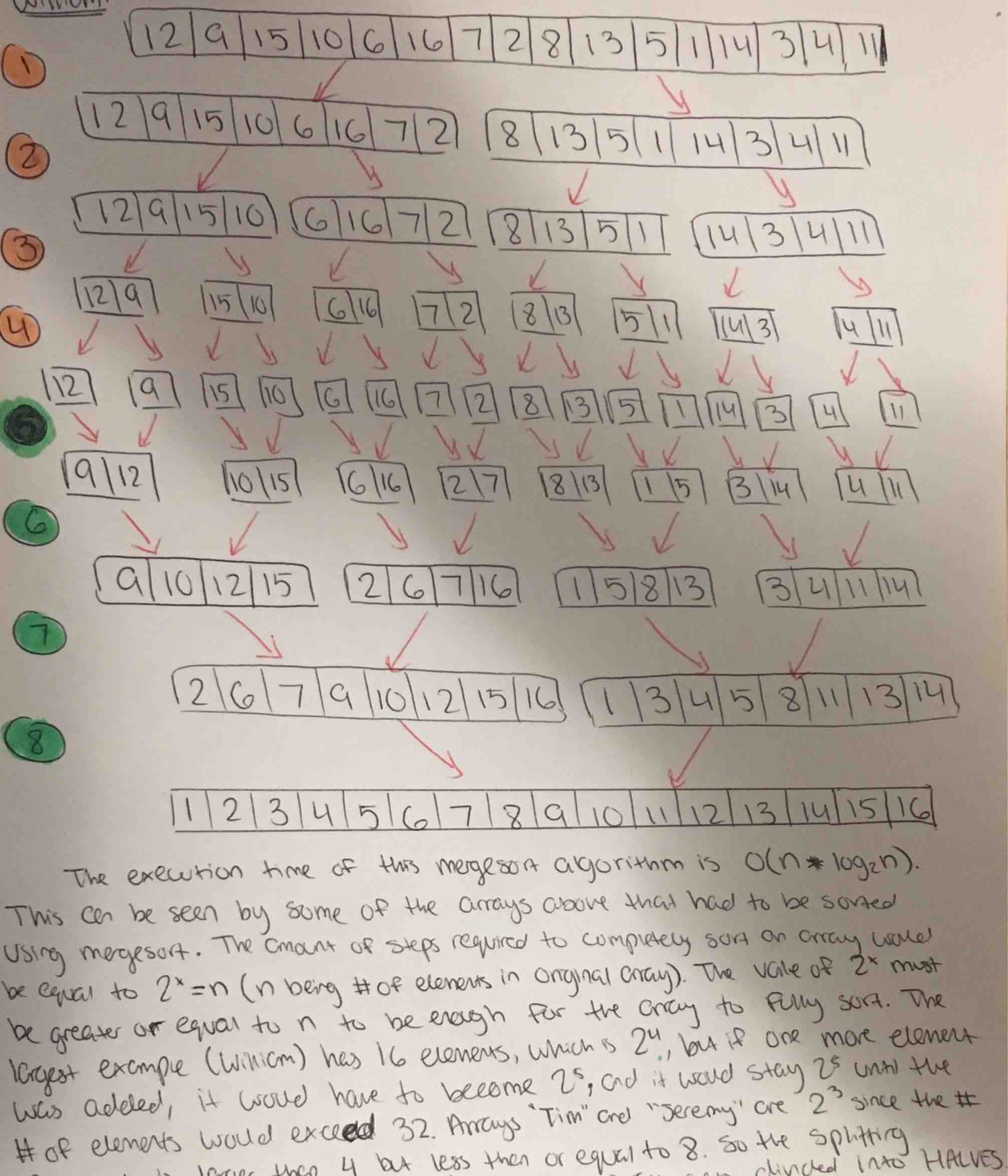


Richard

1
2
3
4
5
6
7
8



William:



The execution time of this mergesort algorithm is $O(n * \log_2 n)$. This can be seen by some of the arrays above that had to be sorted using mergesort. The amount of steps required to completely sort an array would be equal to $2^x = n$ (n being # of elements in original array). The value of 2^x must be greater or equal to n to be enough for the array to fully sort. The largest example (William) has 16 elements, which is 2^4, but if one more element was added, it would have to become 2^5, and it would stay 2^5 until the # of elements would exceed 32. Arrays "Tim" and "Jeremy" are 2^3 since the # of elements is larger than 4 but less than or equal to 8. So the splitting part of the algorithm is $O(\log_2 n)$ since the array gets divided into HALVES. The merging part of the algorithm is just $O(n)$ since it is linear, like discussed earlier. So the resulting execution time is $\boxed{O(n * \log_2 n)}$.