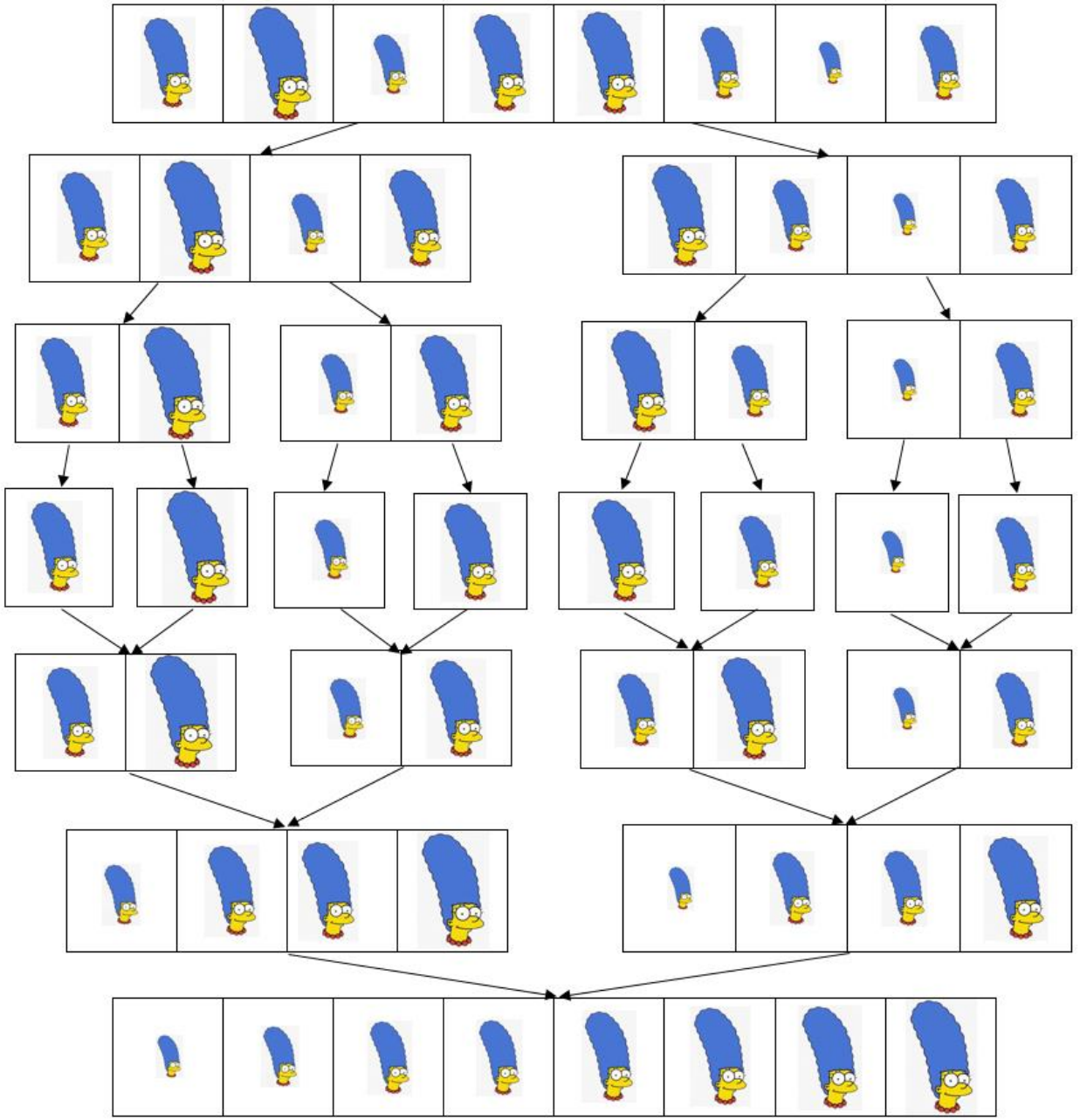
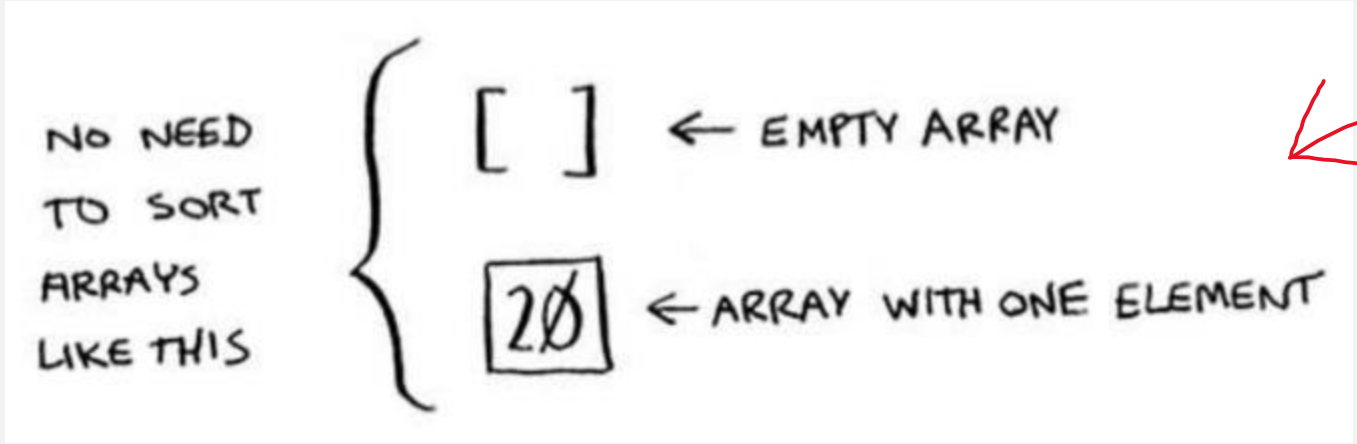
MERGE SORT

Marge Sort

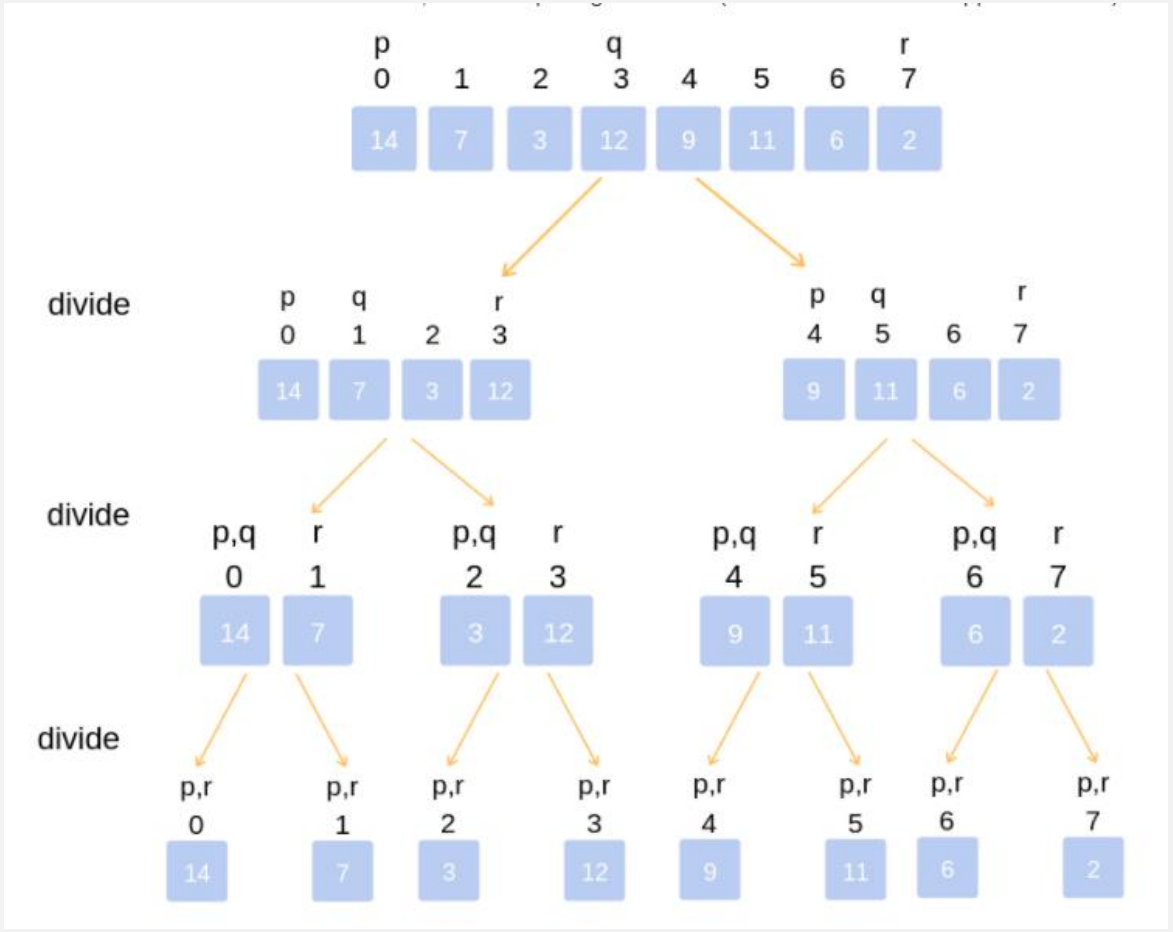


DIVIDE AND CONQUER

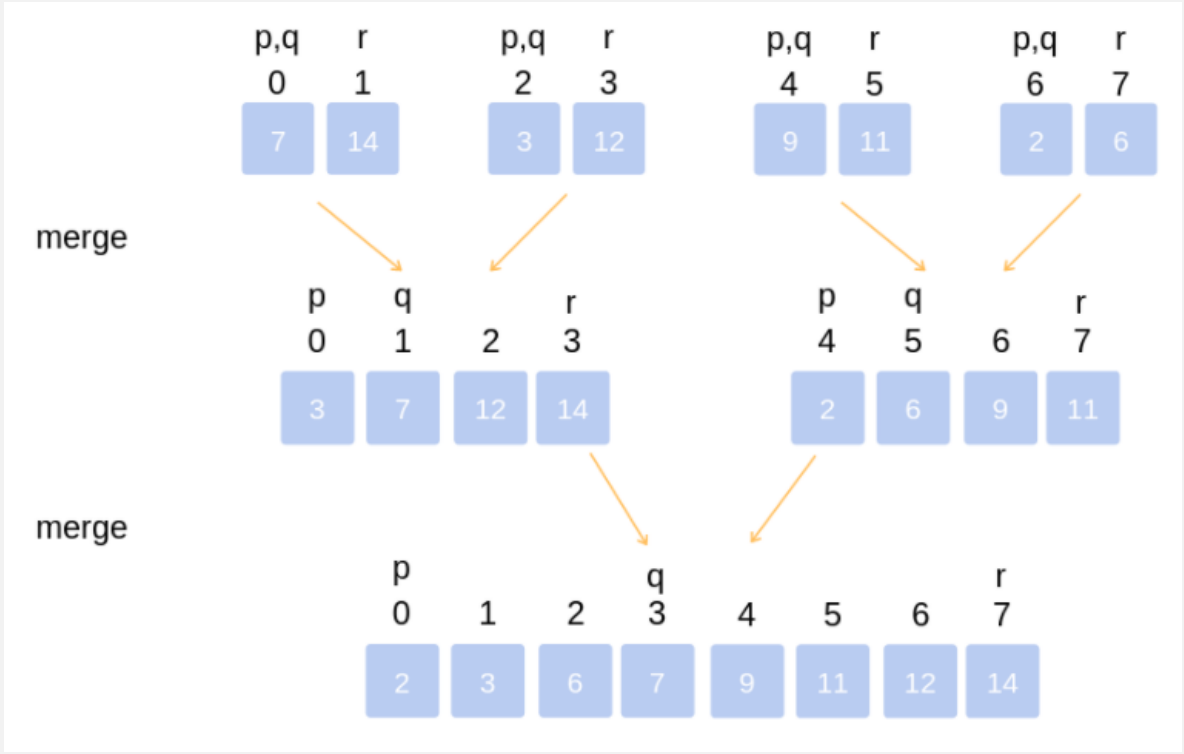


The base case

DIVIDE AND CONQUER

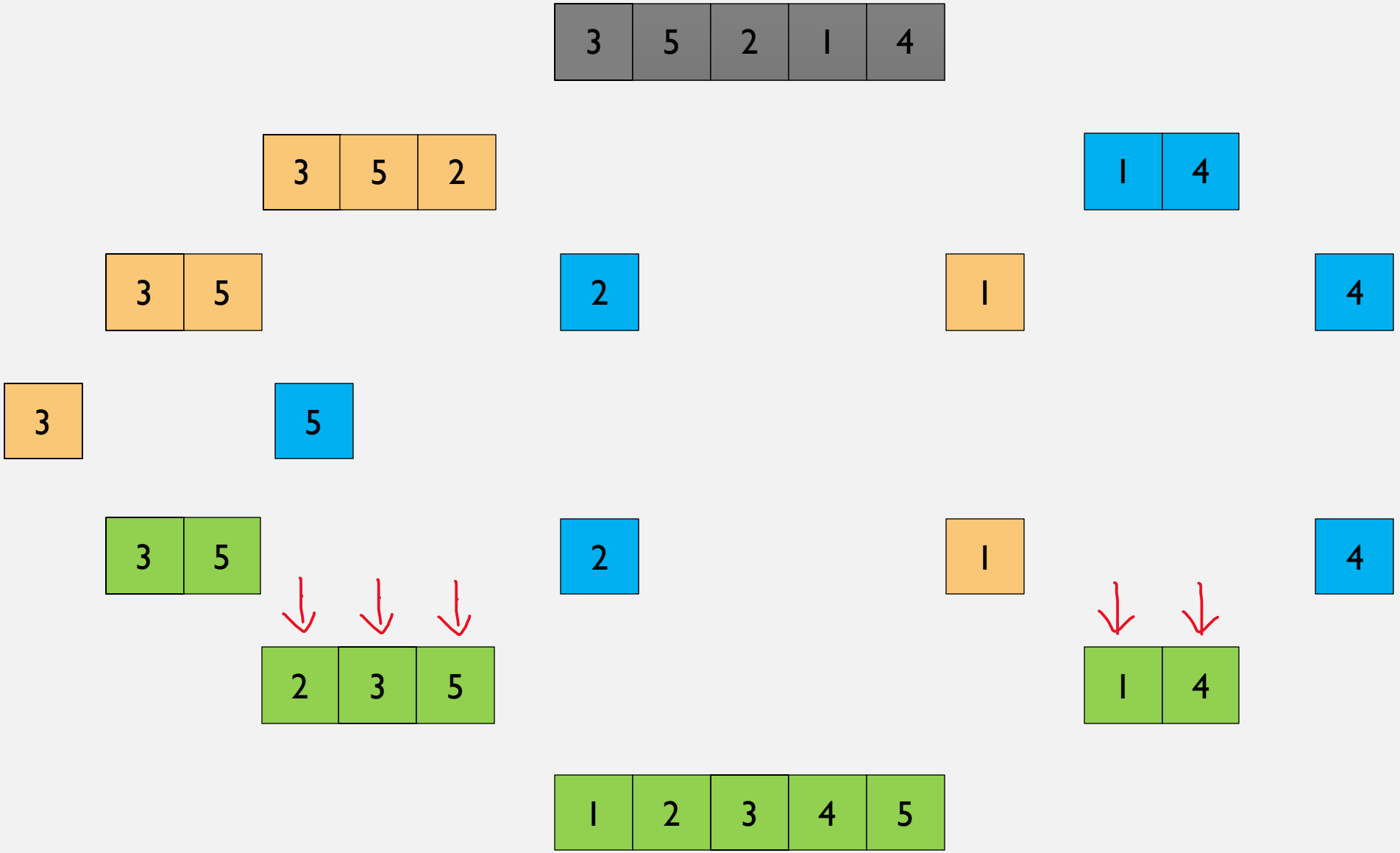


Divide into subarrays



Merge each pair of arrays
(Conquer and Combine)

DIVIDE AND CONQUER APPROACH



https://visualgo.net/en/sorting

"TALK IS CHEAP
SHOW ME THE CODE"

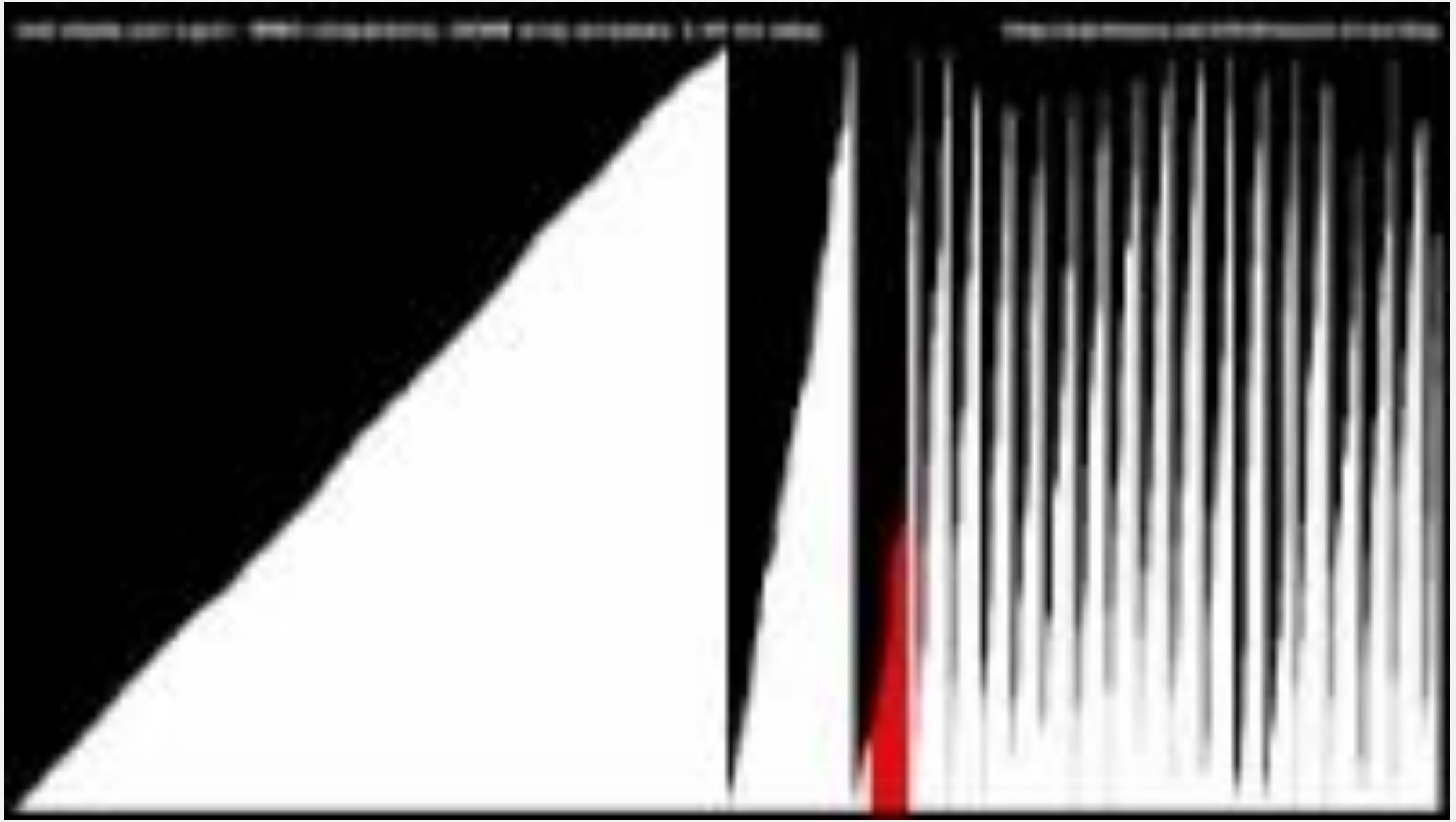
- Linus Torvalds


Cool! But…
How efficient is this?

O(N * LOG(N))

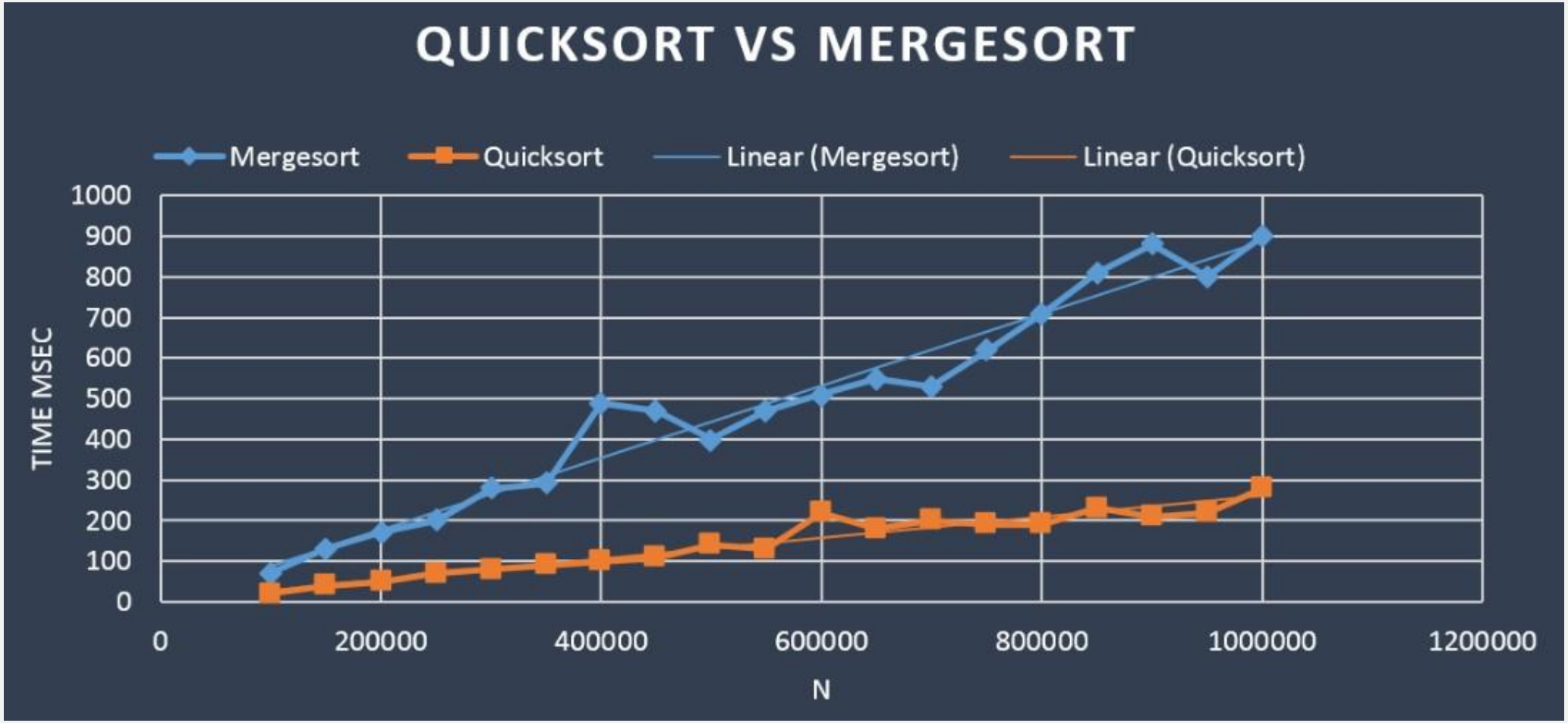
On both AVERAGE and WORST cases!!!!!


*MERGE SORT
I took nlog(n) time to sort
*QUICK SORT
nlog(n) for me too
*HEAP SORT
Hey! nlog(n) for me too!
*BUBBLE SORT
You guys are done sorting???
ifunny.co



MERGE SORT VS QUICKSORT

Sorting algorithm	Average Case	Worst Case
Merge Sort	$O(n * \log(n))$	$O(n * \log(n))$
Quicksort	$O(n * \log(n))$	$O(n^2)$



Source: https://stackoverflow.com/questions/29723648/confusion-about-my-quicksort-algorithm-mergesort-algorithm?rq=1

Quick Sort vs. Merge Sort

- Quick sort
  - **hard division, easy combination**
  - partition in the divide step of the divide-and-conquer framework
  - hence combine step does nothing
- Merge sort
  - **easy division, hard combination**
  - merge in the combine step
  - the divide step in this framework does one simple calculation only

SAVE SPACE

- Code a new version of Merge Sort in which the sorting is done in-place, using a minimal amount of extra arrays to help in the process.