Section 6: Sorting

0. Sorting Hat

Suppose we sort an array of numbers, but it turns out every element of the array is the same, e.g.,{17, 17, 17, ..., 17}. (So, in hindsight, the sorting is useless.)

a) What is the asymptotic running time of **insertion** sort in this case?

O(n) - This is the best case runtime of insertion sort as it only requires one pass through the data. Insertion sort will traverse the array but since each element is not less than the one before it, no extra computations are necessary.

b) What is the asymptotic running time of **selection** sort in this case?

$O(n^2)$ - Selection sort always has n^2 runtime, regardless of the nature of data

c) What is the asymptotic running time of **merge** sort in this case?

$O(n*log(n))$ - Merge sort always has $n*log(n)$ runtime, regardless of the nature of data

d) What is the asymptotic running time of **quick** sort in this case?

$O(n^2)$ - This is the worst case runtime of quick sort. When partitioning, every element is going to fall to the same side of the pivot since they all have the same value which essentially only sorts 1 element per iteration of quicksort, leading to the n^2 runtime.