**Merge Sort: Complexity**

**Base case:** \( T(1) = c \)

\[
T(n) = 2 T(n/2) + n
\]

...  

\( T(n) = O(n \log n) \) (best, worst)

We Want:

\[
\frac{n}{2^k} = 1 \quad \Rightarrow \quad n = 2^k \quad \Rightarrow \quad \log n = k
\]

**QuickSort: Best case complexity**

**Base case:** \( T(1) = c \)

\[
T(n) = 2 T(n/2) + n
\]

...  

\( T(n) = O(n \log n) \)

Same as Mergesort

What is best case? Always chooses a pivot that splits array in half at each step

**QuickSort: Worst case complexity**

\[
T(1) = c
\]

\[
T(n) = n + T(n-1)
\]

...  

\( T(n) = O(n^2) \)

Always chooses WORST pivot – so that one array is empty at each step