- Merge Sort
  - how to analyze the running time?
  - Usually: runtime of a divide and conquer algorithm follows a recursion
  - Let $T(n)$ be the time it takes for merge sort to sort $n$ numbers.

$\text{MergeSort}(a[])$
1. $\text{IF Length}(a) < 2 \text{ THEN RETURN } a.$
2. Partition $a[]$ evenly into two arrays $b[]$, $c[]$.
3. $b[] = \text{MergeSort}(b[])$
4. $c[] = \text{MergeSort}(c[])$
5. $\text{RETURN Merge}(b[], c[])$

$T(n) = T(\frac{n}{2}) + T(\frac{n}{2}) + A \cdot n$

$\uparrow$ step 3 $\uparrow$ step 4 $\uparrow$ step 5

$T(1) = 1$

Goal: solve the recursion
$T(n) = f(n)$

Method 1: Guess and prove by induction. (Better for rigorous proof)

Claim: For $n \geq 2$, $T(n) \leq (A+3)n \log_2 n$

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

Proof: By induction

Base case: $n = 2, 3$, easy to show

Hypothesis: $T(n) \leq (A+3)n \log_2 n$ for $2 \leq n < k$

Want: $T(k) \leq (A+3)k \log_2 k$

Proof: $T(k) = 2T(\frac{k}{2}) + A \cdot k$ (from recursion)

$\leq 2 \cdot (A+3) \cdot \frac{k}{2} \cdot \log_2 \frac{k}{2} + A \cdot k$ (induction hypothesis)

$= (A+3) \cdot k (\log_2 k - 1) + A \cdot k$

$= (A+3) \cdot k \log_2 k - (A+3) \cdot k + A \cdot k$

$\leq (A+3) \cdot k \log_2 k$
- method: recursion tree
  - expand recursion as a tree
  - count: time spent on merging for each layer.
  - take the sum of these costs.

\[
T(n) = \sum_{i=1}^{\log_2 n} \text{merging cost for layer } i
\]

\[
= \sum_{i=1}^{\log_2 n} (\text{#nodes in layer } i \times \text{cost per node})
\]

\[
= \sum_{i=1}^{\log_2 n} 2^{i-1} \times \left( A \cdot \frac{n}{2^{i-1}} \right)
\]

\[
= \sum_{i=1}^{\log_2 n} A \cdot n = A \cdot n \cdot \log_2 n
\]

\[
T(n) = \sum_{i=1}^{\# \text{layers}} \text{merging cost} + \text{base cost} \times \# \text{base cases}
\]

How to get the formula for recursion tree method?

\[
T(n) = 2T\left(\frac{n}{2}\right) + A \cdot n
\]

\[
= 4T\left(\frac{n}{4}\right) + 2 \cdot A \cdot \frac{n}{2} + A \cdot n
\]

\[
= 8T\left(\frac{n}{8}\right) + 4 \cdot A \cdot \frac{n}{4} + 2 \cdot (A \cdot \frac{n}{2}) + A \cdot n
\]

\[
= \ldots \quad (\text{repeat } \approx \log_2 n \text{ times})
\]

\[
= n \cdot T(1) + \sum_{i=1}^{\log_2 n} \frac{n}{2^i} \cdot A \cdot \left(\frac{n}{2^i}\right) + \ldots + 4 \cdot (A \cdot \frac{n}{2}) + 2 \cdot (A \cdot \frac{n}{4}) + A \cdot n
\]

\[
= \sum_{i=1}^{\log_2 n} \text{merge cost for layer } i
\]

\[
= \sum_{i=1}^{\log_2 n} A \cdot n \cdot \log_2 n
\]
\[ = n \cdot 1 + A \cdot n + A \cdot n + \ldots + A \cdot n + A \cdot n + A \cdot n \underbrace{\ldots}_{\text{log}_2 n \text{ terms}}\]

\[ = n + A \cdot n \log_2 n \]

can be ignored in asymptotic analysis.