- analyze running time of divide-and-conquer.
  
  merge-sort(al)
  
  0) base case
  1) split al into bl, cl
  2) merge-sort(bl), merge-sort(cl) recursive call
  3) merge(bl, cl)

  merge cost
  cost of steps 0) 1) 3)
  can be analyzed directly

  recursion cost
  cost of step 2)

  - to analyze the running time
  let T(n) be the running time of the alg.
  for input of size n.
  - analyze merge cost: function of n
  - analyze recursion cost: written as T(k)
    for some k < n
  
  - T(n) = merge cost + recursion cost.

  - analyzing merge-sort
    
    - merge cost 0(n)
    - recursion cost 2 \times T(n/2)

    # recursive calls problem size for each call

    T(n) = 0(n) + 2T(n/2)

  base case T(1) = 0
  T(n) = 2T(n/2) + n

  1) guess-and-verify

  guess: T(n) \leq Cn \log_2 n

  verify: induction

  hypothesis: T(n) \leq Cn \log_2 n
verify: induction

hypothesis: \( T(n) \leq \left\lfloor \frac{Cn \log_2 n}{2} \right\rfloor \)

base case: \( n = 1 \) \( T(1) = 0 = C \cdot \log_2 1 \) (true for every \( C \))

induction: suppose IH is true for all \( k < n \)
will show IH is also true for \( n \)

\[
T(n) = 2T\left(\frac{n}{2}\right) + n \quad \text{(cases recursion)}
\]

\[
\leq 2 \cdot \left\lfloor \frac{C \cdot \frac{n}{2} \log_2 \frac{n}{2}}{2} \right\rfloor + n \quad \text{(IH)}
\]

\[
= 2 \cdot C \cdot \frac{n}{2} \cdot (\log_2 n - 1) + n
\]

want \( T(n) \leq Cn \log_2 n \)

need \( Cn \log_2 n - Cn + n \leq Cn \log_2 n \)

\( C \geq 1 \)

whenever \( C \geq 1 \), we have \( T(n) \leq Cn \log_2 n \)

\( T(n) \leq n \log_2 n \), \( T(n) = O(n \log_2 n) \)

2) recursion tree

- draw a tree of all recursive calls
- node \( \leftrightarrow \) recursive call
- edge \( \leftrightarrow \) one calls the other
- leaf \( \leftrightarrow \) base case

Claim: \( T(n) = \sum \text{merge cost for every node on the recursion tree} \)

for merge sort

\[
T(n) = \sum_{i=0}^{\log_2 n - 1} \text{merge cost for level } i
\]

\[
= \sum_{i=0}^{\log_2 n - 1} 2^i \cdot \left\lfloor \frac{n}{2^i} \right\rfloor
\]

\( \left\lfloor \frac{n}{2^i} \right\rfloor \) nodes at level \( i \)
merge cost for each node at level \( i \)
\[
= \sum_{i=0}^{\Sigma} n = n \log_2 n
\]

- interpret recursion tree method

\[
T(n) = 2T(\frac{n}{2}) + n
\]

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

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

- merge cost for layer 0
- merge cost for layer 1

\[
= \sum_{i=0}^{\text{\# layers}} \text{merge cost for layer } i
\]