MERGE Sort, Divide & Conquer, dealing with recurrence relations

Divide problem into 2 smaller instances

Conquer (= solve) the smaller problems (Mergesort)

Combine (= merge) the solutions
Merging 2 sorted arrays:

In general: copy to output, increment, smallest remaining element found with 1 comparison.

Smallest element is at leftmost position of A or B.

A: 1 2 3 5 9 14 16 20

B: 4 6 10 11 15 19 25 31

output: 1 2 3 4
Mergesort time for \( n \) elements: \( T(n) \)

1) Divide \( \Theta(1) \)

2) Conquer \( \Theta(1) + 2 \cdot T(\frac{n}{2}) \)

3) Merge \( \Theta(n) \)

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

\[
T(1) = \Theta(1)
\]

Actually \( T(n) = T(\frac{n}{2^j}) + T\left(\frac{n}{2^j}\right) + \Theta(n) \) \( \rightarrow \) easy to deal with
How to solve \( T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n) \)

\( T(n) = 2 \cdot T\left(\frac{n}{2}\right) + c \cdot n \) ← must do this → \( = c_2 \)

Recursion tree:

\[
\begin{array}{c}
\text{c.n} \\
T\left(\frac{n}{2}\right) \quad T\left(\frac{n}{2}\right)
\end{array}
\]

\[
\begin{array}{c}
\text{c.n} \\
T\left(\frac{n}{4}\right) \quad T\left(\frac{n}{4}\right) \\
\text{c.n/2} \quad 
\end{array}
\]

\[
\begin{array}{c}
\text{c.n/4} \\
T\left(\frac{n}{8}\right) \quad T\left(\frac{n}{8}\right) \\
\text{c.n/8} \quad 
\end{array}
\]

\[
\begin{array}{c}
\text{c.n/8} \\
\vdots
\end{array}
\]

\( n \) leaves: \( c_2 \quad c_2 \quad c_2 \ldots \)

Total:
\[
c_2 n + cn \log_2 n = \Theta(n \log n)
\]
The recursion tree method relies on noticing a pattern and for it to be formal one must prove that the pattern holds not just for a few levels.

If that’s not easy to do, use substitution and induction...
How to solve \( T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n) \) by substitution (induction)

\( T(n) = 2 \cdot T\left(\frac{n}{2}\right) + cn \)  \( \left\uparrow \text{must do this} \right\uparrow \)

1) You need to have a guess for the answer. \( \left\uparrow \text{we control this} \right\uparrow \)

\( T(n) \leq \Theta(n \log n) \)  \( \rightarrow \Theta(n \log n) \)

2) Inductive hypothesis: for all \( k < n \), \( T(k) \leq d \cdot k \log k \)

3) Substitute: \( T(n) \leq 2 \cdot d \cdot \frac{n}{2} \log \frac{n}{2} + cn \) (using \( k = \frac{n}{2} \))

\[
= dn \log n - dn \log 2 + cn
= dn \log n - (dn - cn)
\leq dn \log n \quad \text{if} \quad d \geq c
\]

3) Algebra:

\( \left\{ \begin{array}{l}
\leq dn \log n \quad \text{... if} \quad d \geq c
\end{array} \right. \)  \( \square \)  \( \text{(base case omitted)} \)
More observations:  
\[ T(n) = 2 \cdot T\left(\frac{n}{2}\right) + c \cdot n \]

- For an upper bound, you must get the desired form or less. (exactly: with the same leading constant)

- Like any inductive proof, if it doesn't work, that doesn't imply it's not true.

- For mergesort specifically, the constant \( c \) comes from the merge step...and this ends up as the leading constant.  
  \[ T(n) \leq c n \log n \]
  (Speedup of mergesort is directly proportional to speedup of merge)
\[ T(n) = 4T\left(\frac{n}{2}\right) + n \]

\[ T(1) = 1 \]

\[ T(2) = 4T(1) + 2 = 4 \cdot 1 + 2 = 6 \]

\[ T(4) = 4T(2) + 4 = 4 \cdot 6 + 4 = 28 \]

\[ T(8) = 4T(4) + 8 = 4 \cdot 28 + 8 = 120 \]

\[ T(16) = 4T(8) + 16 = 4 \cdot 120 + 16 = 496 \]

\[ T(32) = 4T(16) + 32 = 4 \cdot 496 + 32 = 2016 \] : starts looking like \(2n^2\)

2nd term less and less significant
\[ T(n) = 4T\left(\frac{n}{2}\right) + n \quad T(1) = 1 \]

What should we guess?

It looked quadratic. Not really convincing though. Let's be cautious.

Try \( O(n^3) \): \( T(n) \leq cn^3 \)

Hypothesis: \( T(k) \leq ck^3 \) for \( k < n \)

Substitute: \( T(n) \leq 4 \cdot c \cdot \left(\frac{n}{2}\right)^3 + n \)

Algebra:

\[
\begin{align*}
= \frac{1}{2} cn^3 + n \\
= cn^3 - \frac{1}{2} cn^3 + n \\
= cn^3 - \left(\frac{1}{2} cn^3 - n\right)
\end{align*}
\]

desired form

\[
\begin{align*}
\geq 0 \text{ if } c > 1 & \& n > 2 \\
\text{for all } n \geq 0
\end{align*}
\]

but now you must handle a larger base case

Base case: \( T(1) = 1 \leq c \cdot 1^3 \) \( \checkmark \)
We wanted to show $T(n) = 4T\left(\frac{n}{2}\right) + n = O(n^3)$.

Why use hypothesis $T(k) \leq ck^3$ instead of $T(k) = O(k^3)$?

e.g., $T(n) = 4T\left(\frac{n}{2}\right) + n$

$$= 4 \cdot O\left(\left(\frac{n}{2}\right)^3\right) + n = 4 \cdot O\left(\frac{1}{8} n^3\right) + n = O(n^3)$$

Let's show $T(n) = n$ is $O(1)$

Base case: $T(1) = O(1)$

Hypothesis: $T(k) = O(1)$ for all $k < n$

$\therefore T(n-1) = n-1 = O(1)$

$T(n) = n = n-1 + 1 = O(1) + 1 = O(1)$
$$T(n) = 4T\left(\frac{n}{2}\right) + n \quad \text{/} \quad T(1) = 1 \quad \text{/} \quad \text{We proved } O(n^3). \text{ Let's try } O(n^2)$$

$$T(n) \overset{?}{\leq} cn^2$$

Hypothesis: $$T(k) \leq ck^2 \quad \text{for} \quad k < n$$

Substitute: $$T(n) \leq 4 \cdot c \cdot \left(\frac{n}{2}\right)^2 + n$$

Algebra:

$$= cn^2 + n \quad \rightarrow \text{failed to get } \leq \text{ desired form}$$

$$= cn^2 + \frac{1}{n}n^2 = \left(c + \frac{1}{n}\right)n^2 \quad \left\{\begin{array}{l}
\text{so close to } cn^2 \\
\text{but not good enough}
\end{array}\right.$$
\[ T(n) = 4T\left( \frac{n}{2} \right) + n \quad \text{%} \quad T(1) = 1 \quad \text{%} \quad \text{We proved } O(n^3). \quad \text{We want } O(n^2) \]

Hypothesis: \( T(k) \leq ck^2 \) *failed.* New hypothesis: \( T(k) \leq ck^2 - dk \)

Substitute: \( T(n) \leq 4 \cdot (c\left( \frac{n}{2} \right)^2 - d\left( \frac{n}{2} \right)) + n \)

Algebra:

\[ = cn^2 - 2dn + n \]

\[ = cn^2 - dn - dn + n \]

\[ = cn^2 - dn - (dn - n) \]

\[ \geq 0 \text{ if } d \geq 1 \]

So can \( c \) be anything? No.

Base case: \( T(1) = 1 \leq c \cdot 1^2 - d \cdot 1 \) if \( c > d \)

Making the hypothesis stronger often works.

See my notes on induction.
$T(n) = 4T\left(\frac{n}{2}\right) + n$
$T(n) = 4T\left(\frac{n}{2}\right) + n$

Geometric series

Sum = $2n^2$

Leaf level $\sim \log_2 n$

$2^{\log_2 n} \cdot n = n^2$

$\sum = O(n^2)$
\[ T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2 \]

Tree is not balanced

\[ T(n) \leq n^2 \left[ 1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \left(\frac{5}{16}\right)^3 + \cdots + \left(\frac{5}{16}\right)^\infty \right] \]

\[ < n^2 \cdot \left[ 1 + \frac{1}{2} + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^3 + \cdots \right] = 2n^2 \]

\[ \therefore \left(\frac{5}{16}\right)^k \]

Exaggerate & simplify

Pretend the pattern holds forever