# Solving Recurrence Equations

In MergeSort, we used a technique called recursion tree method to get the time complexity bound $(n \log n)$ assuming $n$ to be a power of 2.

For arbitrary $n$, we modified the algorithm slightly by adding dummy $\infty$ to the input to make it to be of size, the next higher power of 2.
In the end discard the $\infty$'s.
Works in $O(n \log n)$ time for all $n$.

Suppose we do not wish to modify the algorithm, but want to analyse the time complexity of the original algorithm. Then we need to solve recurrence eqn,

$$T(n) = T(\lfloor \tfrac{n}{2} \rfloor) + T(\lceil \tfrac{n}{2} \rceil) + \underbrace{\Theta(n)}$$

Today, we see how to solve this eqn.

$c_1' n \leq \cdots \leq c_1 n \quad c_1', c_1 > 0$

1. This eqn splits into two equations, one can be used to give an upper bound and the other may be used to give a lower bound on $T(n)$

2. Since we are interested only in asymptotic form of $T(n)$ we may replace some finitely many values of $T(n)$ bounded by a constant $c_0$

$$T(n) \leq \begin{cases} T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + c_1 n & \text{if } n > n_0 \\ c_0 & \text{if } n \leq n_0 \end{cases}$$

$c_0, c_1 > 0$

$n_0 \geq 1$

$\longrightarrow$ 1

# Substitution Method

Assumes that we have a form of $T(n)$ and we wish to verify it.

Guess and verify.

Claim $T(n) \leq d_2 n \log n + d_1 n$ for all $n$

$d_2 > 0$

We prove this inductively

Base case $n \leq n_0$

$T(n) = c_0$. If we choose $d_1 \geq c_0$

then $c_0 \leq d_2 n \log n + d_1 n$

## Induction step $\quad n > n_0$

$$T(n) \le T(\lfloor \tfrac{n}{2} \rfloor) + T(\lceil \tfrac{n}{2} \rceil) + c_1 n$$

(By I.H.)

$$\le \underline{d_2 \lfloor \tfrac{n}{2} \rfloor \log \lfloor \tfrac{n}{2} \rfloor} + d_1 \lfloor \tfrac{n}{2} \rfloor$$
$$d_2 \lceil \tfrac{n}{2} \rceil \log \lceil \tfrac{n}{2} \rceil + d_1 \lceil \tfrac{n}{2} \rceil + c_1 n$$

$$\le d_2 \lfloor \tfrac{n}{2} \rfloor \log \lceil \tfrac{n}{2} \rceil + d_1 (\lfloor \tfrac{n}{2} \rfloor + \lceil \tfrac{n}{2} \rceil) + c_1 n$$
$$d_2 \lceil \tfrac{n}{2} \rceil \log \lceil \tfrac{n}{2} \rceil$$

$$\le d_2 (\log \lceil \tfrac{n}{2} \rceil) . n + d_1 n + c_1 n$$

$\lceil \tfrac{n}{2} \rceil \le \tfrac{2}{3} n \quad$ for $n \ge 6 \quad$ [Exercise]

$$T(n) \le d_2 n \log \tfrac{2}{3} n + d_1 n + c_1 n$$

$$= d_2 n \log n + d_2 n \log \tfrac{2}{3} + d_1 n + c_1 n$$

We need to show that $T(n) \le d_2 n \log n + d_1 n$

For this it suffices to show that

$$d_2 n \log \tfrac{2}{3} + c_1 n \le 0 \quad \text{for all } n > n_0$$

$$\Longleftrightarrow \quad -d_2 n \log \tfrac{2}{3} \ge c_1 n$$

$$\Longleftrightarrow \quad -d_2 \log \tfrac{2}{3} \ge c_1$$

$$\Longleftrightarrow \quad d_2 \ge \boxed{\frac{c_1}{-\log \frac{2}{3}}}$$

$c_1 > 0$, $-\log \frac{2}{3} > 0$

Choosing $d_2 = \dfrac{c_1}{-\log \frac{2}{3}}$ $d_1 = c_0$

ensures the induction step for all $n \geq 6$ $\square$
and the base case.

If we take $n_0 = 6$

then $T(n) \leq d_2 n \log n + d_1 n$

$O(n \log n)$

# Another Example

$$T(n) = 4T\left(\frac{n}{3}\right) + \theta(n)$$

It is known that $T(n) = \theta\left(n^{\log_3 4}\right)$

[Follows by master thm, to be covered later]

We ignored here $\lceil\ \rceil$, $\lfloor\ \rfloor$. We assume that $n$ is a power of 3.

**Claim** $T(n) \leq d\, n^{\log_3 4}$

$$T(n) \leq \begin{cases} 4T\left(\frac{n}{3}\right) + c_1 n & \text{if } n > n_0 \\ c_0 & \text{if } n \leq n_0 \end{cases}$$

$c_0, c_1 > 0$

$n_0 \geq 1$

We proceed by induction to prove our claim.

<u>Base Case</u> $n \leq n_0$

$T(n) = c_0 \leq d\, n^{\log_3 4}$ if $d \geq c_0$

<u>Induction Step</u>

$$T(n) \leq 4T\left(\frac{n}{3}\right) + c_1 n$$

$$\leq 4\, d\left(\frac{n}{3}\right)^{\log_3 4} + c_1 n$$

$$= \cancel{4}\, d\, \frac{n^{\log_3 4}}{\cancel{3^{\log_3 4}}} + c_1 n$$

$$= d\, n^{\log_3 4} + c_1 n$$

We need to show that $\leq d\, n^{\log_3 4}$ for all $n > n_0$

This is <u>FALSE!</u>

We need to strength our induction hypothesis, for induction step to go through

$$T(n) \leq d_2 n^{\log_3 4} + d_1 n \quad (4) \qquad (d_1 < 0)$$

$$\boxed{\text{Importance of subtracting lower order terms}}$$

Induction Step

$$T(n) \leq 4T\left(\frac{n}{3}\right) + c_1 n$$

$$\leq 4\left[d_2\left(\frac{n}{3}\right)^{\log_3 4} + d_1 \frac{n}{3}\right] + c_1 n$$

$$= d_2 n^{\log_3 4} + \frac{4}{3} d_1 n + c_1 n$$

We need to show that

$$d_2 n^{\log_3 4} + \frac{4}{3} d_1 n + c_1 n \leq d_2 n^{\log_3 4} + d_1 n$$

It suffices to show that

$$\frac{d_1 n}{3} + c_1 n \leq 0 \quad \text{for all } n > n_0$$

$$\iff \frac{d_1}{3} \leq -c_1$$

$$\iff d_1 \leq -3c_1$$

For base case,

$$c_0 \leq d_2 n^{\log_3 4} + d_1 n \quad \text{for } n \leq n_0$$

$$d_2 = c_0 + |d_1|$$

$$c_0 \leq \underbrace{c_0 + |d_1|} + d_1$$

$$\leq d_2 + d_1 \leq d_2 n^{\log_3 4} + d_1 n$$

$n_0 = 1$, $d_1 = -3c_1$, $d_2 = c_0 + 3c_1$ works

$\square$

To apply Substitution method, we need an initial guess.

We may combine this technique with the recursion tree method, recursion tree method provides initial guess.

Verification may be done by Substitution method.