# Divide and-Conquer - I



- In divide and Conquer we solve a problem recursively, applying three steps at each level of recursion.

1. **Divide** the problem into a no. of subproblem that are smaller instances of the same problem.

2. **Conquer** the subproblem by solving them recursively. When the subproblem sizes are small, solve them straightforward.

3. **Combine** the solutions to the subproblems into the solution for the original problem.

- Recursive case - when subproblems are large enough to solve recursively.

- Base Case - When subproblems become small enough that no longer recurse (recursion "bottom out").

- Solve subproblems that are not same as original problem in Combine Steps.

## Recurrences

- It is an equation or inequality that describe a function in terms of its value on smaller inputs.

- eg Merge Sort

$$T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 2T(n/2) + \Theta(n) & \text{if } n>1 \end{cases}$$

$$= \Theta(n \log n)$$

- Subproblems are not necessarily constrainted to being a constant fraction of original size. E.g linear search

Create one Subproblem Containing only one element fewer than the Original problem.

$$T(n) = T(n-1) + \Theta(1)$$

- There are 3 methods for Solving recurrences
- Substitution method
- Recursion - tree method
- Master theorm method.

## ① Substitution Method

- The substitution method for solving recurrences Comprises 2 Steps
1. Guess the form of the Solution.
2. Use mathematical induction to find Constants and Show that the Solution works

- It is powerful but Can be applied only in Cases when it is easy to guess the form of the answer.

- Can be used to established either upper or lower bound.

eg 1 $T(n) = 2T(\lfloor n/2 \rfloor) + n$

We guess the sol $T(n) = O(n \log n)$

Substitution method require to prove that $T(n) \le Cn \log n$

We Assume that bound holds for all positive $m < n$ particular $m = \lfloor n/2 \rfloor$ yielding

$$T(n) \le 2(C\lfloor n/2 \rfloor \log(\lfloor n/2 \rfloor)) + n$$
$$\le Cn \log(n/2) + n$$
$$= Cn\log n - Cn\log 2 + n = Cn\log n - n(C\log_2 - 1)$$
$$= Cn\log n$$
$$\le Cn\log n.$$

when $C \ge 1$

(Ajay Rawat)

classmate
Date
Page

**eg 2** Consider the recurrence $T(n) = 2T\left(\lfloor n/2 \rfloor + 16\right) + n$

Show that it is asymptotically bound by $O(n \log n)$

- $T(n) = O(n\log n)$ we have to show that for some constant $c$

$$T(n) \le cn\log n$$

Put this in given recurrence equation

$$T(n) \le 2\left[c\left(\lfloor n/2 \rfloor + 16\right)\log\left(\lfloor n/2 \rfloor + 16\right)\right] + n$$

$$= cn\log(n/2) + 32 + n$$

$$= cn\log n - cn\log 2 + 32 + n$$

$$= cn\log n - cn + 32 + n$$

$$= cn\log n - (c-1)n + 32$$

$$\le cn\log n \quad \text{for } (c \ge 1)$$

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

**eg 3**

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left(\lfloor n/2 \rfloor\right) + n & n > 1 \end{cases}$$ find asymptotic bound on $T$

- We guess the solution is $O(n\log n)$. Thus for a constant $c$

$$T(n) \le cn\log n$$

$$T(n) \le 2c\lfloor n/2 \rfloor \log\lfloor n/2 \rfloor + n$$

$$\le cn\log n - cn\log 2 + n$$

$$= cn\log n - n(c\log_2 2 - 1)$$

$$\le cn\log n \quad \forall\ c \ge 1$$

**Making a good guess**

1. If a recurrence is similar to one we have seen before then guessing a similar solution is reasonable.

2. Another way is to prove loose upper and lower bounds on the recurrence and then reduce the range of uncertainty.

3. In a guess, when math doesn't seem to work out in the induction, we revise the guess by subtracting a lower order term then

maths often goes through.

4. Changing variables. - Sometimes, a little algebraic manipulation can make a unknown recurrence similar to one we have seen before.

<u>Example</u> $T(n) = 2T(\sqrt{n}) + \log n$. Solve it by changing variable.

Suppose $m = \log_2 n \Rightarrow n = 2^m$

$$n^{1/2} = 2^{m/2} \Rightarrow \sqrt{n} \Rightarrow 2^{m/2}$$

Put value we get

$$T(2^m) = 2T(2^{m/2}) + m$$

Again $S(m) = T(2^m)$

$$S(m) = 2S\left(\frac{m}{2}\right) + m$$

We know this recurrence has solution

$$S(m) = O(m \log m)$$

Substitute the value of m.

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

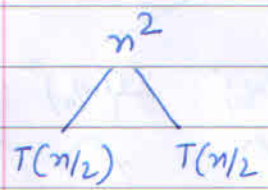
$$= O(m \log m) = O(\log n \log \log n)$$

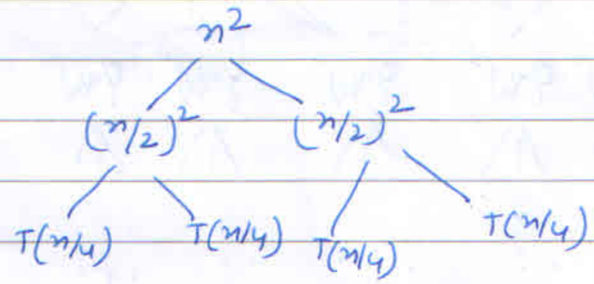
# Divide and Conquer

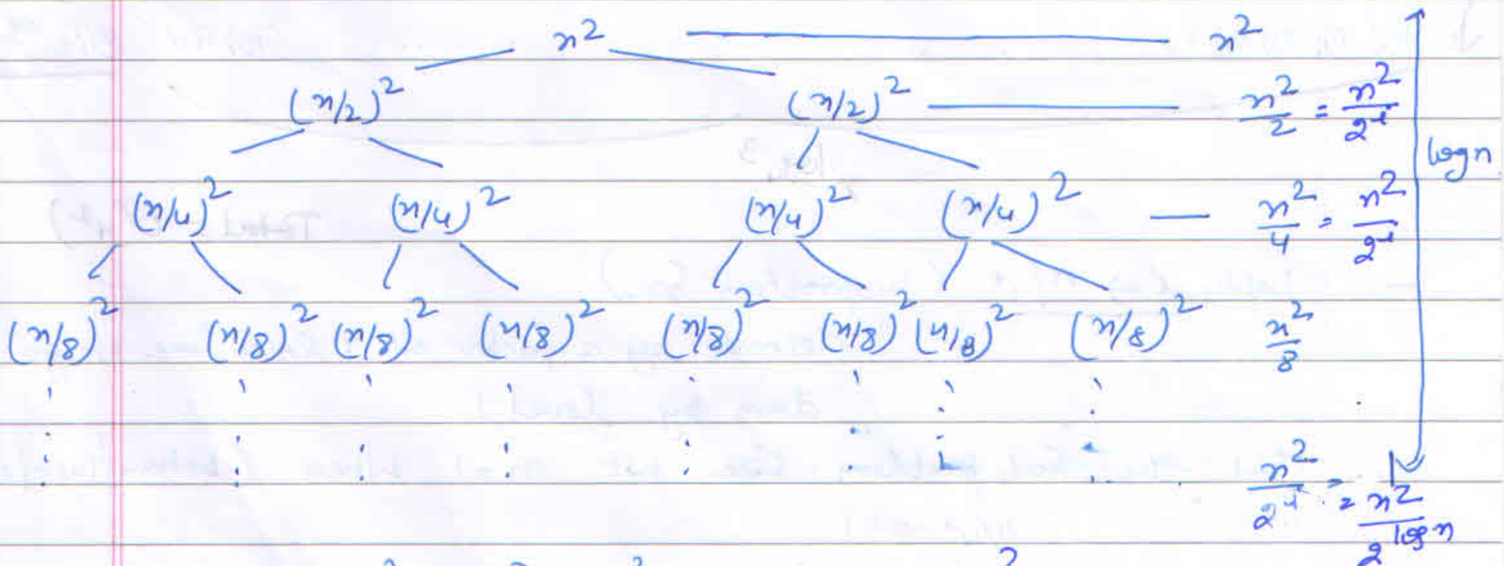
## 2. Recursion Tree

- Each node represents the cost of a single sub problem.
- Sum the costs within each level of the tree to obtain a set of per-level costs.
- Then sum all the per-level cost to determine the total cost of all level of the recursion.

<u>Example</u> - Consider $T(n) = 2T\left(\frac{n}{2}\right) + n^2$, obtain asymptotic bound using recursion tree method.



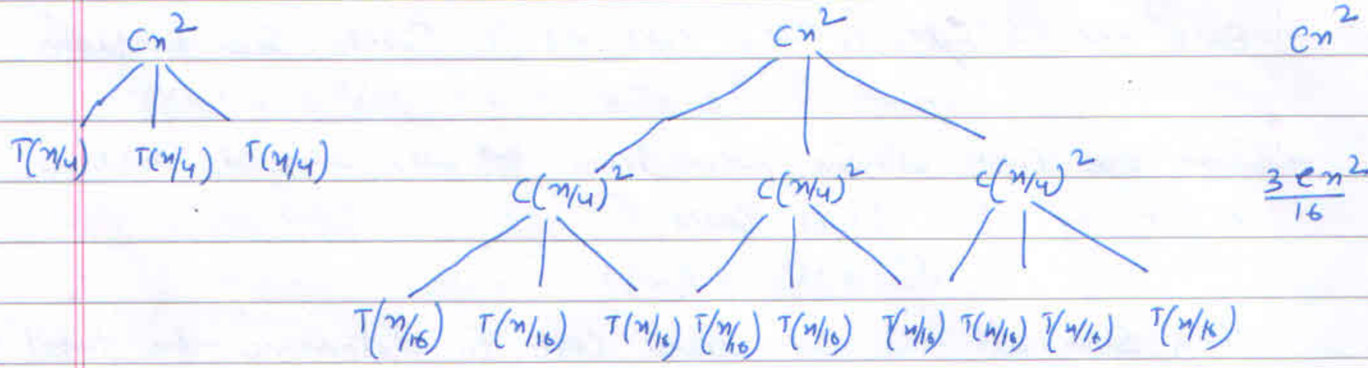



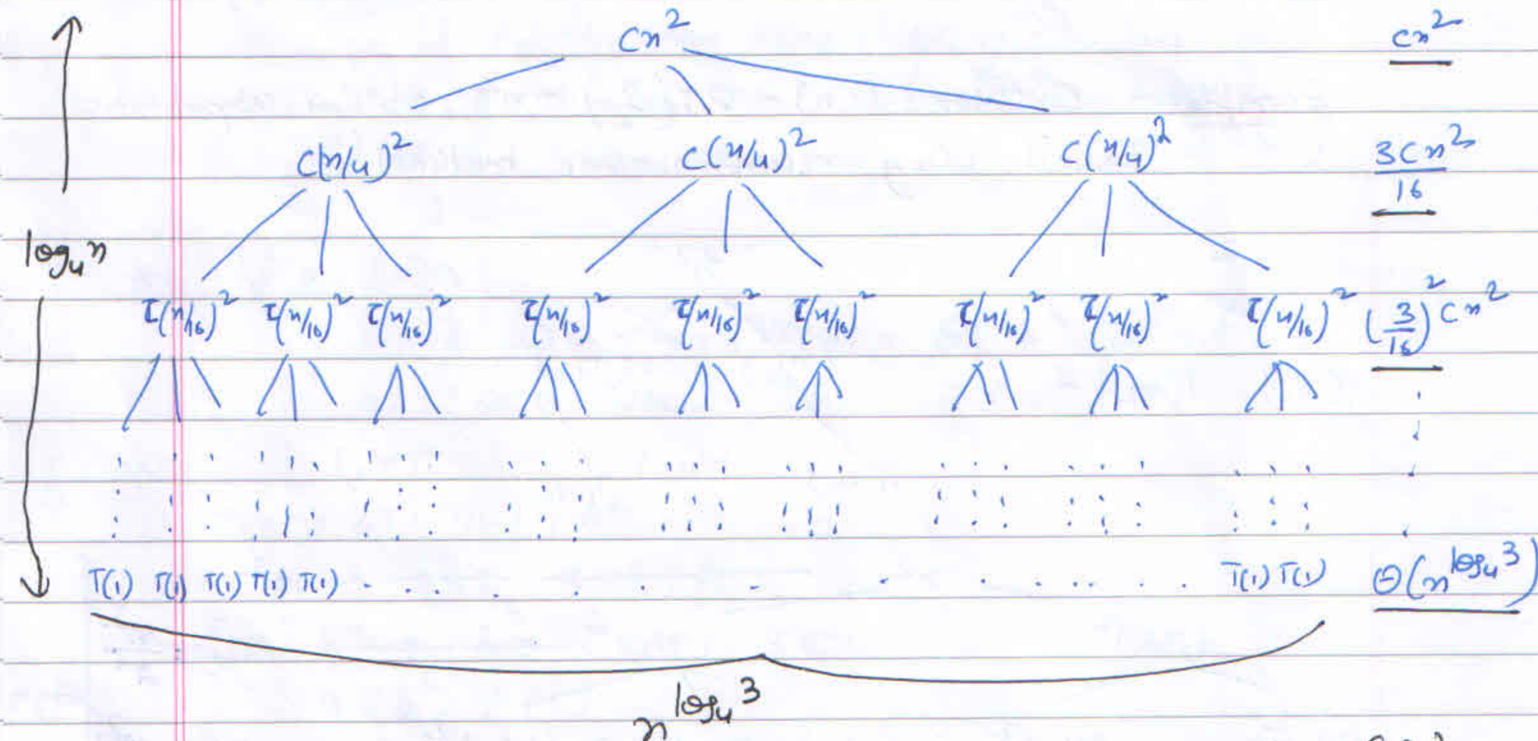

$$T(n) = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \frac{n^2}{8} - - - - + \frac{n^2}{2^{\log n}}$$

$$T(n) = \Theta(n^2)$$

<u>Depth</u> - At each level $i$ subproblem size $= \frac{n}{2^i}$

Example - $T(n) = 3T(n/4) + cn^2$




Total = $O(n^2)$

- Depth $\underline{i \Rightarrow n/4^i}$ (Subproblem Size)
  decrease by a factor of 4 each time we go down by level 1

∴ Thus the Sub problem size hit $n=1$ When (bottom/last/level)

$$n/4^i = 1$$

$$\underline{i = \log_4 n}$$

- Number of nodes at $i = 3^i$
- Each node at depth $i$ has cost $= c(n/4^i)^2$
- So ~~total~~ node cost at depth $i = 3^i c(n/4^i)^2 = \underline{(3/16)^i cn^2}$

(because of, 3)

- Bottom level at depth $\log_4 n$ has $3^{\log_4 n} = n^{\log_4 3}$
each contributing cost $T(1)$, total cost $= \Theta(n^{\log_4 3})$
When we add all these we get $\underline{\underline{\Theta(n^2)}}$

1)- Solve the following recurrence

$$T(n) = 4T\left(\lfloor n/2 \rfloor\right) + n \quad \text{Using recursion tree.}$$

2) Consider the following recurrence

$$T(n) = T\left(n/3\right) + T\left(\frac{2n}{3}\right) + n$$

hint:

$$\begin{tikzcd}[column sep=small] & n \arrow[dl, dash] \arrow[dr, dash] & \\ T(n/3) & & T(2n/3) \end{tikzcd}$$

3) **Master Theorm**

let $T(n)$ be defined on the non-negative integers by the relurrence.

$T(n) = aT(n/b) + \Theta(n^k \log^p n)$ where

$a \geq 1$, $b > 1$, $k \geq 0$ and $p$ is a real number

1) if $a > b^k$ then $T(n) = \Theta(n^{\log_b a})$

2) if $a = b^k$

a) if $p > -1$ then $T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$

b) If $p = -1$ then $T(n) = \Theta(n^{\log_b a} \log \log n)$

c) If $p < -1$ then $T(n) = \Theta(n^{\log_b a})$

3) if $a < b^k$

a) if $p \geq 0$ then $T(n) = \Theta(n^k \log^p n)$

b) if $p < 0$ then $T(n) = O(n^k)$

Examples

1) $T(n) = 3T(n/2) + n^2$

$a = 3, b = 2, k = 2, p = 0$

$a = b^k \Rightarrow 3 = 2^2 = 4$, $3 < 4$

$a < b^k$ (3a)

$T(n) = \Theta(n^2 \log^0 n) = \Theta(n^2)$

2) $T(n) = 4T(n/2) + n^2$

$a = 4, b = 2, k = 2, p = 0$

$4 = 4$ (Case 2a)

$T(n) = \Theta(n^{\log_2 4} \log^1 n) = (n^2 \log n)$

3) $T(n) = T(n/2) + n^2$

$a = 1, b = 2, k = 2, p = 0$

$1 < 4$ (Case 3a)

$T(n) = \Theta(n^2)$

4) $T(n) = 2^n T(n/2) + n^n$

$a = 2^n$ it should be constant

So Cannot be Solved with Master Theorm.

5) $T(n) = 16T(n/4) + n$

$a = 16$, $b = 4$, $k = 1$, $p = 0$

$16 > 4$ (Case - 1)

$T(n) = \Theta(n^{\log_4 16}) = \Theta(n^2)$

6) $T(n) = 2T(n/2) + n \log n$

$a = 2$, $b = 2$, $k = 1$, $p = 1$

$2 = 2$ (Case 2a)

$T(n) = \Theta(n^{\log_b a} \log^{p+1} n) = \Theta(n^1 \log^2 n) = \Theta(n \log^2 n)$

7) $T(n) = 2T(n/2) + n/\log n$

$a = 2$, $b = 2$, $k = 1$, $p = -1$

$2 = 2$ (Case 2b)

$T(n) = \Theta(n^1 \log \log n) = \Theta(n \log^2 n)$

8) $T(n) = 2T(n/4) + n^{0.51}$

$a = 2$, $b = 4$, $k = 0.51$, $p = 0$, $2 < 4^{0.51}$ (Case 3a)

$T(n) = \Theta(n^{0.51})$

9) $T(n) = 0.5T(n/2) + 1/n$

$a = 0.5$, a should be $\geq 1$

So Cannot be Solved by Master theorm

10) $T(n) = 6T(n/3) + n^2 \log n$

$a = 6$, $b = 3$, $k = 2$, $p = 1$

$6 < 3^2$ (Case 3a)

$T(n) = \Theta(n^2 \log^1 n) = \Theta(n^2 \log n)$

classmate
Date
Page

11) $T(n) = 64T(n/8) - n^2 \log n$
because of -ve sign cannot be solved.

12) $T(n) = 7T(n/3) + n^2$
$a = 7, \; b = 3, \; k = 2, \; p = 0$
$7 < 9$ (Case 3 a)
$T(n) = \Theta(n^2)$

13) $T(n) = 4T(n/2) + \log n$
$a = 4, \; b = 2, \; K = 0, \; p = 1$
$4 > 2^0$ (Case - I)
$T(n) = \Theta(n^{\log_2 4}) = \Theta(n^2)$

14) $T(n) = \sqrt{2}\, T(n/2) + \log n$
$a = \sqrt{2}, \; b = 2, \; K = 0, \; p = 1$
$\sqrt{2} > 2^0$ (Case - I)
$T(n) = \Theta(n^{\log_2 \sqrt{2}}) = \Theta(\sqrt{n})$

15) $T(n) = 2T(n/2) + \sqrt{n}$
$a = 2, \; b = 2, \; k = 1/2, \; p = 0$
$2 > 2^{1/2}$ (Case - I)
$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n)$

16) $T(n) = 3T(n/2) + n$
$a = 3, \; b = 2, \; k = 1, \; p = 0$
$3 > 2^1$ (Case - I)
$T(n) = \Theta(n^{\log_2 3})$

17) $T(n) = 3T(n/3) + \sqrt{n}$
$a = 3, \; b = 3, \; k = 1/2, \; p = 0$
$3 > 3^{1/2}$ (Case - I)
$T(n) = \Theta(n^{\log_3 3}) = \Theta(n)$

18) $T(n) = 4T(n/2) + cn$

$a = 4, \; b = 2, \; k = 1, \; p = 0$

$4 > 2^1$ (Case - I)

$T(n) = \Theta(n^{\log_2 4}) = \Theta(n^2)$

19) $T(n) = 3T(n/4) + n\log n$

$a = 3, \; b = 4, \; k = 1, \; p = 1$

$3 < 4^1$ (Case 3a)

$T(n) = \Theta(n^{1} \log^2 n) = \Theta(n\log n)$