Recurrence: $T(n) = aT(n/b) + f(n)$

- a: #subproblems
- n/b: size of the subproblem
- $f(n)$: combine/divide time

$T(1) = O(1)$ — work done at leaf

Total Value: $T(n) = \sum_{i=0}^{\log_b n - 1} a^i f(n/b^i) + a^{\log_b n} \cdot O(1)$

- $\log_b n - 1$: levels except leaves
- $i = 0$: root
- a^i: # nodes in that level
- $f(n/b^i)$: function on that size input
- $a^{\log_b n}$: #leaves
- $O(1)$: work done at each leaf.

Work done on each level i: $f(n/b^i)$ ← level $\log_b n$

Num nodes on each level: a^i

Num leaves = $a^{\log_b n}$ (# nodes last level)

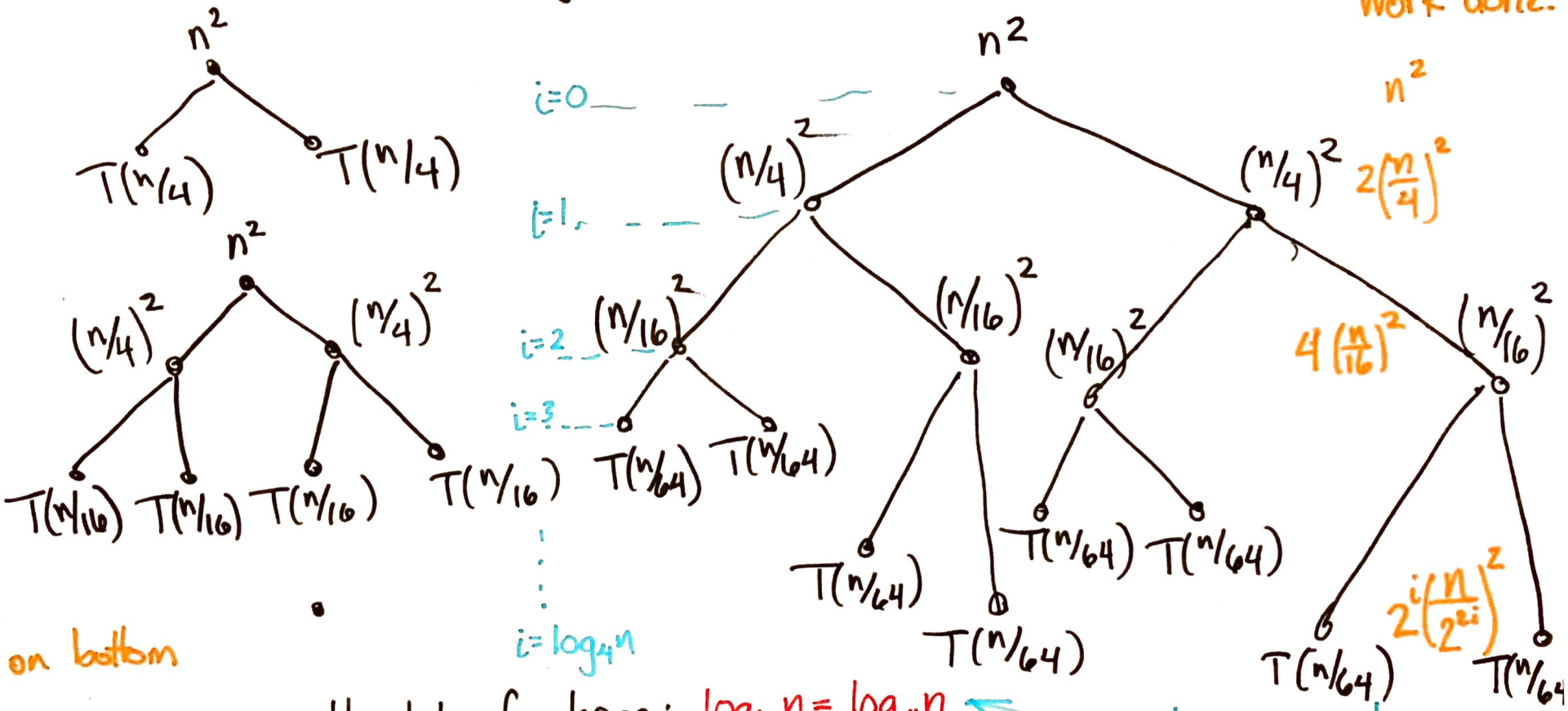
Height of tree: $\log_b n$

Levels in tree: height + 1

Leaves in tree: $a^{\log_b n}$

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

$a = 2$
$b = 4$



Work done on bottom level:
leaves × O(1)
$= 2^{\log_4 n}$

Work done in rest of tree:

$$\text{height} \uparrow \quad \sum_{i=0}^{\log_4 n - 1} 2^i \left(\frac{n}{2^i}\right)^2$$

Amt of work done on each level.

Height of tree: $\log_b n = \log_4 n$ ← how many times do we divide the input by 4 to get to the bottom?

Number of leaves: $a^{\log_b n} = 2^{\log_4 n}$ ← 2 subproblems, so 2^i nodes on each level i.

Nodes on each level: 2^i

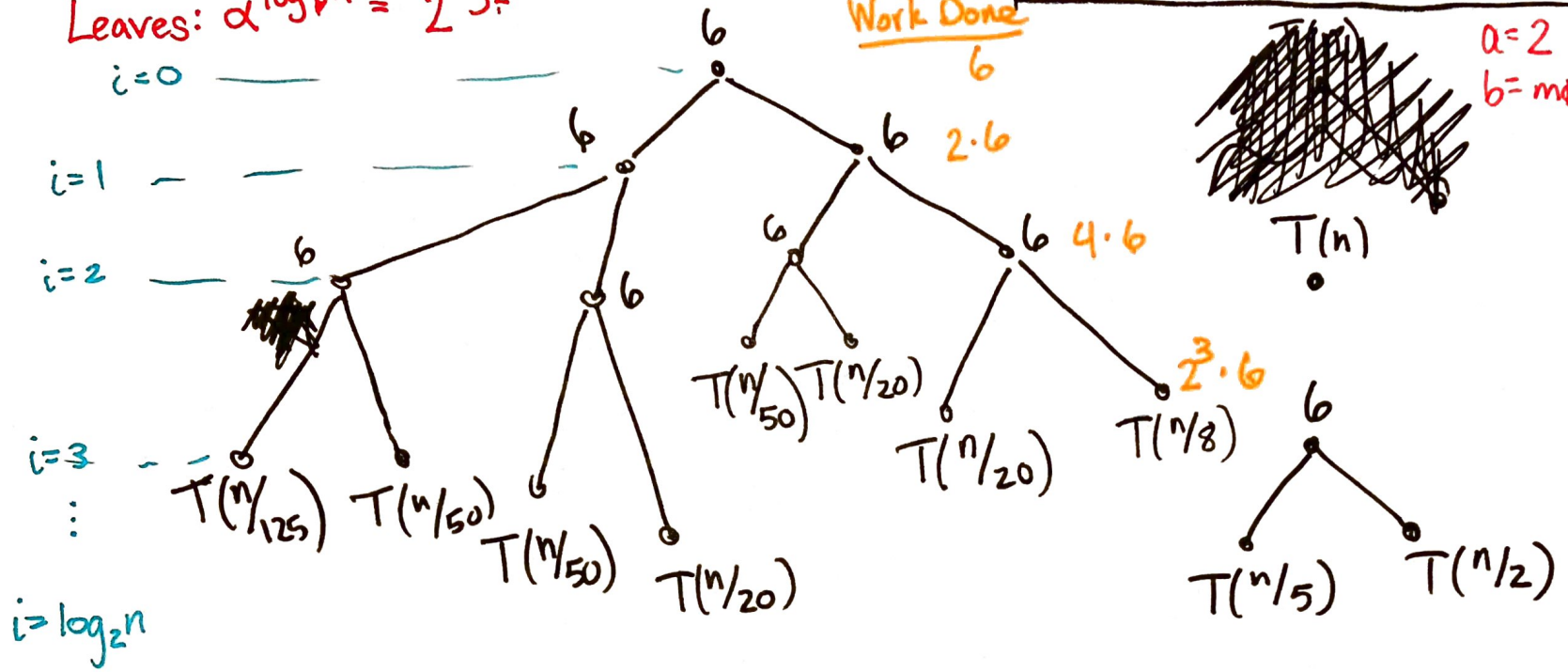
Number of levels: height + 1

Levels: Height +1

Height: $\log_\beta n = \log_2 n$

Leaves: $\alpha^{\log_\beta n} = 2^{\log_2 n}$

$T(n) = T(n/5) + T(n/2) + 6$ (2)



Height of tree: $\log_b n, = \log_2 n$

b = size of subproblem

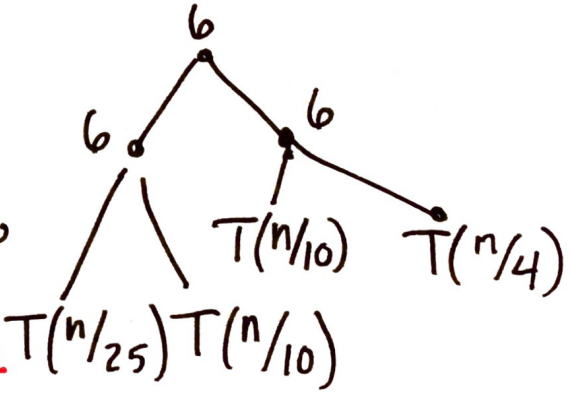
Choose the biggest subproblem, or smallest divisor. (i.e., max(5,2))

Work done on bottom:

$n \times O(1) = n$

Work done in rest of tree:

$\sum_{i=0}^{\log_2 n - 1} 2^i \cdot 6$



Number of leaves: $n^{\log_b a} = n^{\log_2 2} = n$

Work done on each level: 6 · # nodes at that level

#nodes = 2^i, where i = level, starting at i=0 at the root.

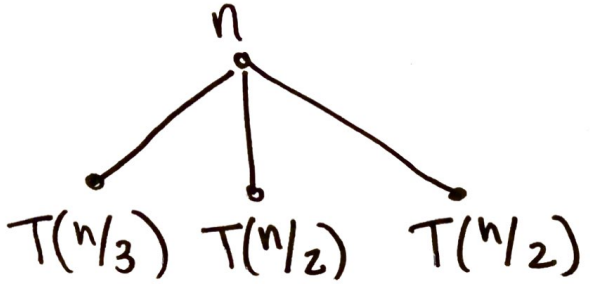
2 because we split into 2 nodes at each lev

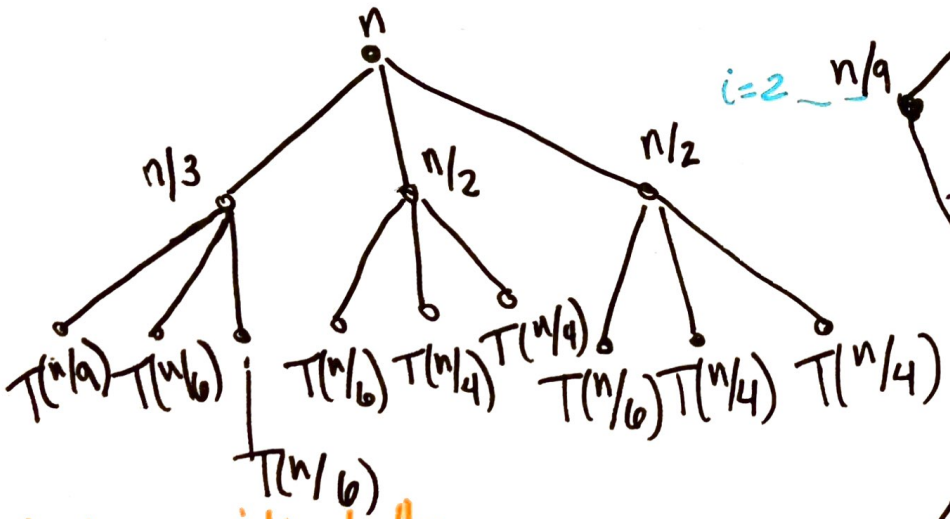
$$T(n) = T(n/3) + 2T(n/2) + n$$

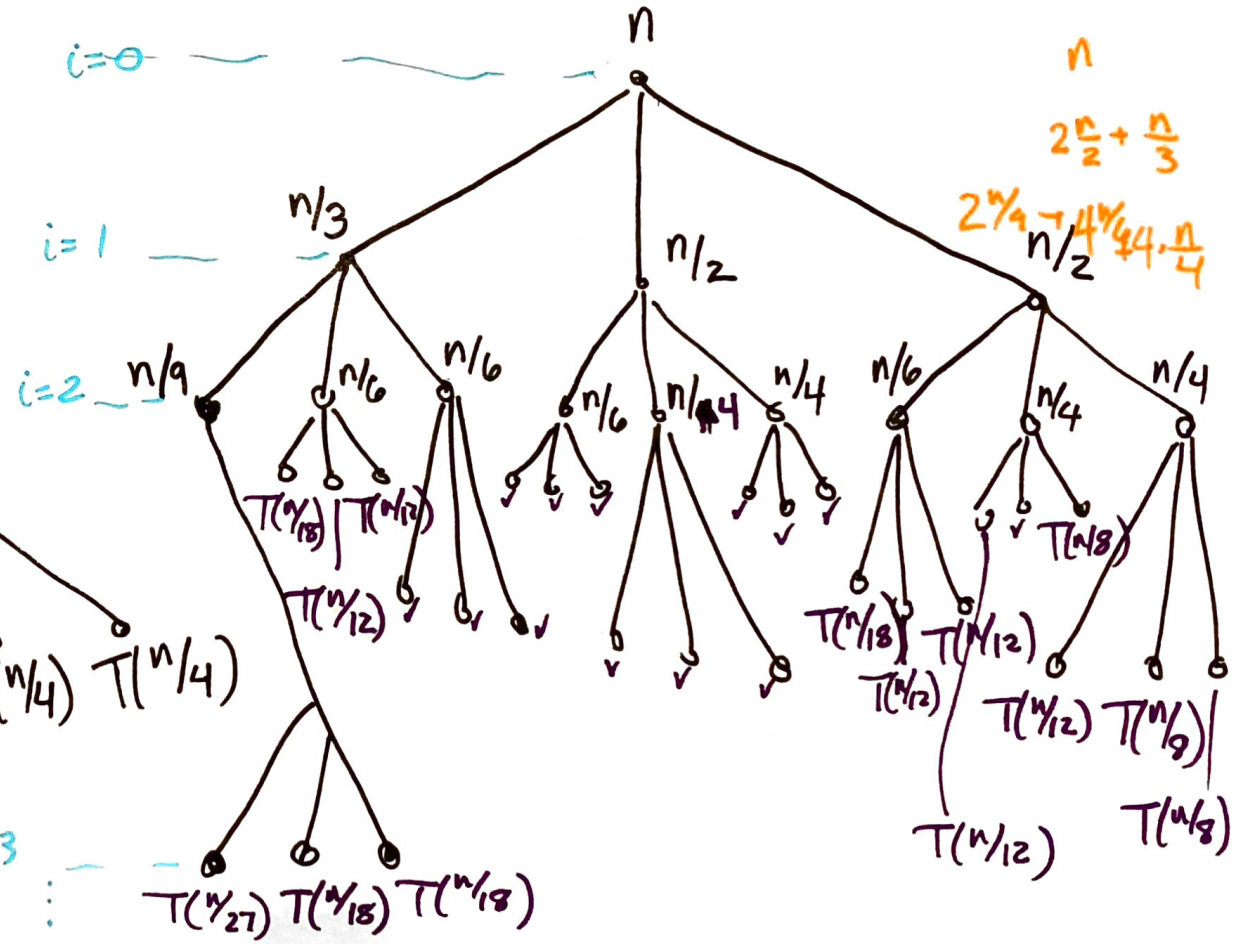
$a = 3$ (# subproblems) 3

$b = \min(2, 3) = 2$

T(n)







Work done on the bottom:

$3^{\log_2 n}$

Work done on the rest of the tree:

$$\sum_{i=0}^{\log n - 1} O(n) \Rightarrow \sum_{i=1}^{\log n} O(n)$$

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

Height of tree: $\log_2 n = \lg n$

Number of leaves: $3^{\log_2 n}$

Nodes on each level: 3^i