

CSSE 473 – Day 4

MICHAEL WOLLOWSKI

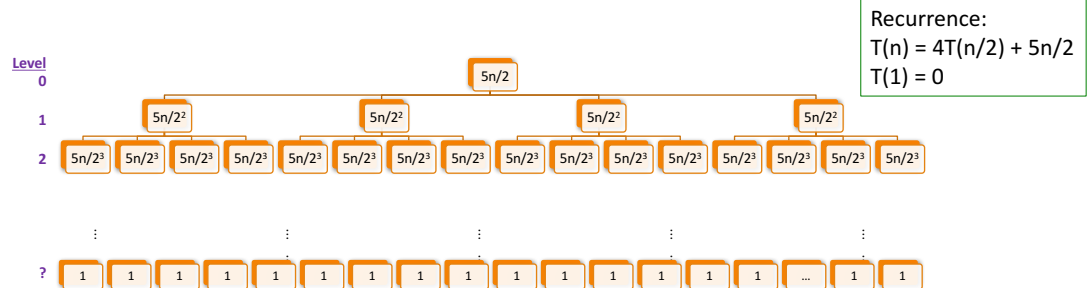
Master Theorem

Let $T(n)$ be a monotonically increasing function that satisfies:

- $T(n) = a T(n/b) + f(n)$
- $T(1) = c$
- where $a \geq 1$, $b \geq 2$. If $f(n) \in \Theta(n^d)$ where $d \geq 0$, then:

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Recurrence Tree



- How many nodes at level i ?
- How much work at level i ?
- Index of final level?

$$4^i$$

$$4^i \left(\frac{5n}{2^{i+1}} \right) = 2^{i-1} 5n$$

$$\log_2 n = k$$

Node: single instance of recursive method
 Node label: time of *non-recursive* work at that node
 Children: recursive calls!