Method of Unrolling: Another example

- Consider the following inductively defined function:

  $T(1) = 1$

  $\forall n \geq 2,\ \ T(n) = 2T(n-1) + 3$

- Let's try the method of unrolling. Suppose n is very large. Then

$$
\begin{aligned}
T(n) &= 2T(n-1) + 3 \quad \text{(by definition)} \\
&= 2(2T(n-2) + 3) + 3 = 2^2T(n-2) + 2\cdot3 + 3 \\
&= 2^2(2T(n-3) + 3) + 2\cdot3 + 3 = 2^3T(n-3) + 2^2\cdot3 + 2\cdot3 + 3 \\
&\ldots \\
&= 2^kT(n-k) + 3\left(\sum_{i=0}^{k-1} 2^i\right)
\end{aligned}
$$

The base case occurs when $k = n - 1$, and hence

$$T(n) = 2^{n-1}\cdot1 + 3\left(\sum_{i=0}^{n-2} 2^i\right) = 2^{n-1} + 3\left(2^{n-1} - 1\right) = 2^{n+1} - 3 \quad \text{which is } O(2^n)$$

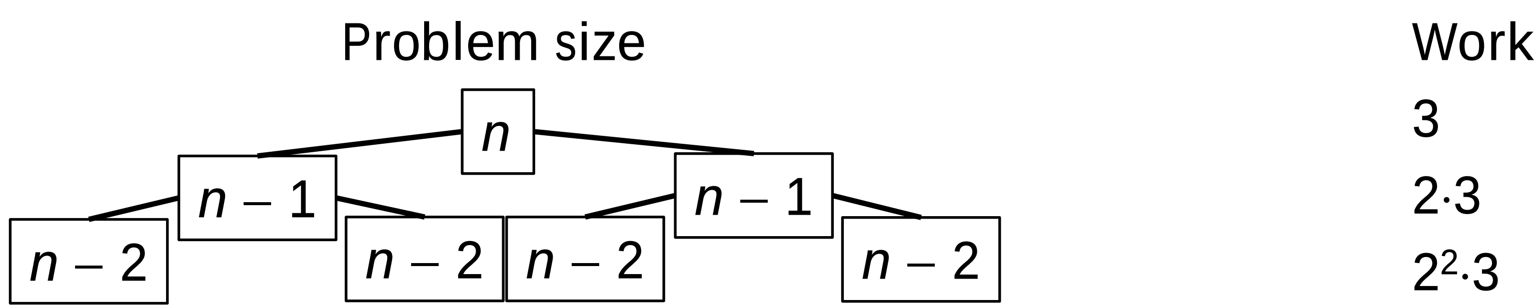
Method of Trees

- Let's consider the same inductively defined function:

  $T(1) = 1$

  $\forall n \geq 2, \; T(n) = 2T(n-1) + 3$

- The inductive step says: The problem $T(n)$ breaks down into 2 smaller problems $T(n-1)$, after doing 3 units of work. We can draw a picture:



- We have a *binary tree* where the work at level 0 (top) is 3, the work at level 1 is $2 \cdot 3$, the work at level 2 is $2^2 \cdot 3$, etc. In general, the work at level k is $2^k \cdot 3$. The problem shrinks to size 1 at the *leaves*, when $k = n-1$

- Thus $T(n)$ = sum of all the work = $2^{n-1} + 3\left(\sum_{i=0}^{n-2} 2^i\right)$ as before

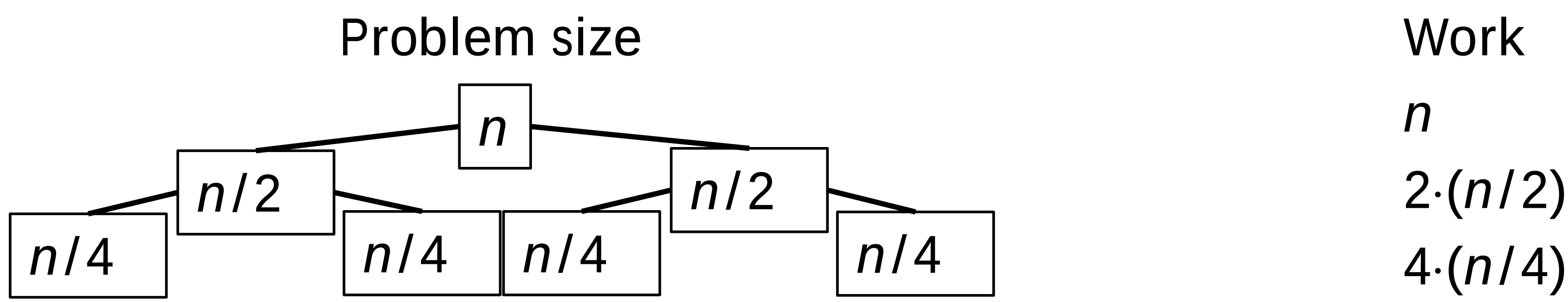
Method of Trees: Another example

- Consider the inductively defined function:

  $T(1) = 1$

  $\forall n \geq 2, \; T(n) = 2T(n/2) + n$

- The inductive step says: The problem $T(n)$ breaks down into 2 smaller problems $T(n/2)$, after doing n units of work. We can draw a picture:



- The work at level 0 (top) is n, the work at level 1 is $2 \cdot (n/2) = n$, the work at level 2 is $4 \cdot (n/4) = n$, etc. In general, the work at level k is n. The problem shrinks to size 1 at the *leaves*, when $k = \log_2 n$

- Thus $T(n)$ = sum of all the work = $n \log_2 n$

Method of Trees: Yet another example

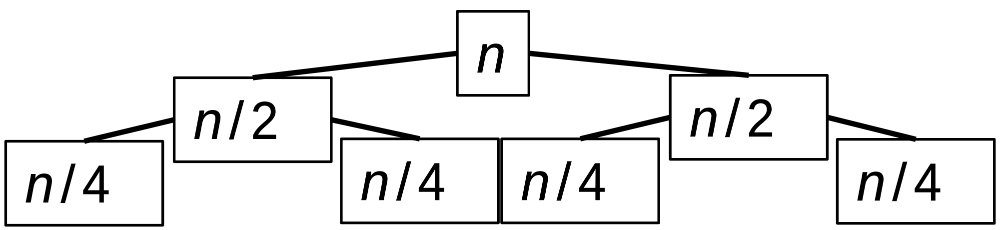
- Consider the inductively defined function:

  $T(1) = 1$

  $\forall n \geq 2, \; T(n) = 2T(n/2) + n^2$

- The inductive step says: The problem $T(n)$ breaks down into 2 smaller problems $T(n/2)$, after doing n^2 units of work. We can draw a picture:

Problem size



Work

n^2

$2\cdot(n/2)^2$

$4\cdot(n/4)^2$

- The work at level 0 (top) is n^2, the work at level 1 is $2\cdot(n/2)^2 = n^2/2$, the work at level 2 is $4\cdot(n/4)^2 = n^2/4$, etc. In general, the work at level k is $n^2/2^k$. The problem shrinks to size 1 at the *leaves*, when $k = \log_2 n$

- Thus $T(n) = n^2\left(\sum_{i=0}^{\log n} \frac{1}{2^i}\right) = n^2\left(2 - \frac{1}{2^{\log n}}\right) = n^2\left(2 - \frac{1}{n}\right)$ which is $O(n^2)$