Recursion Trees

example: base case: $S(1) = C$ (work for base case)

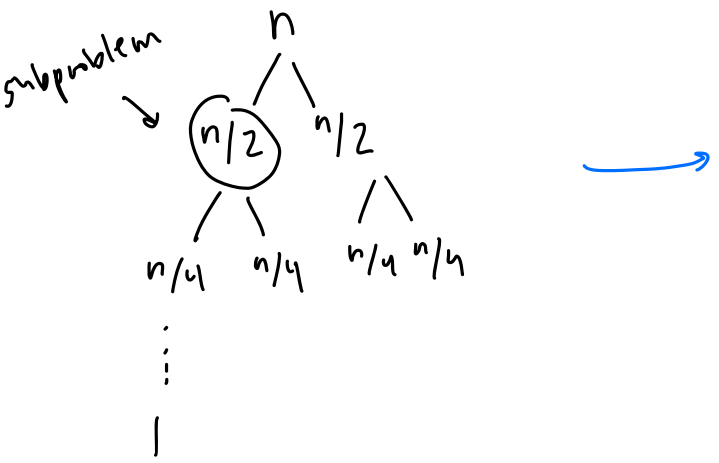
$S(n) = 2S(n/2) + n \quad \forall n \geq 2$ ("extra work")

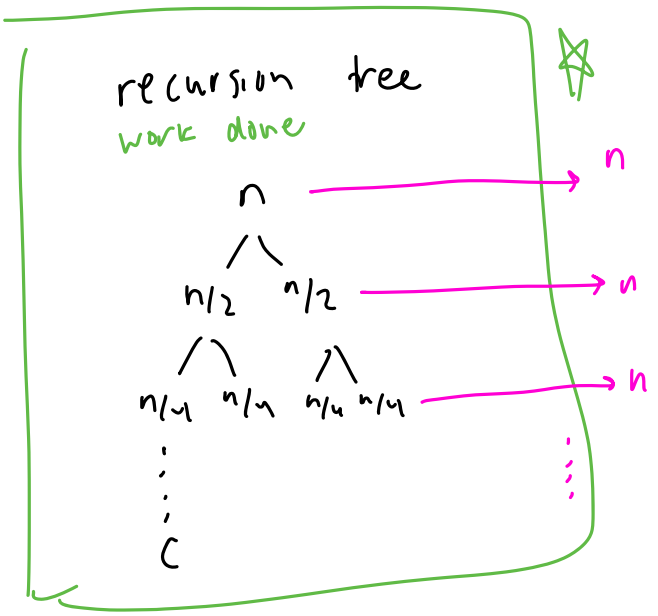
$S(n)$: input of size n

$2S(n/2)$: calls itself twice on input of $n/2$

tree represents size of input

not recursion tree





Question: what is the sum of all values in recursion tree?

internal nodes: $n \cdot$ (# non-leaf levels) = work per level · # non-leaf levels

↓

# of times divide n by 2 to get 1.

$\log_2 n$

internal nodes: $n \log_2 n$

leaves: C (# leaves)

↓

$2^{\log_2 n} = n$

$C \cdot n$

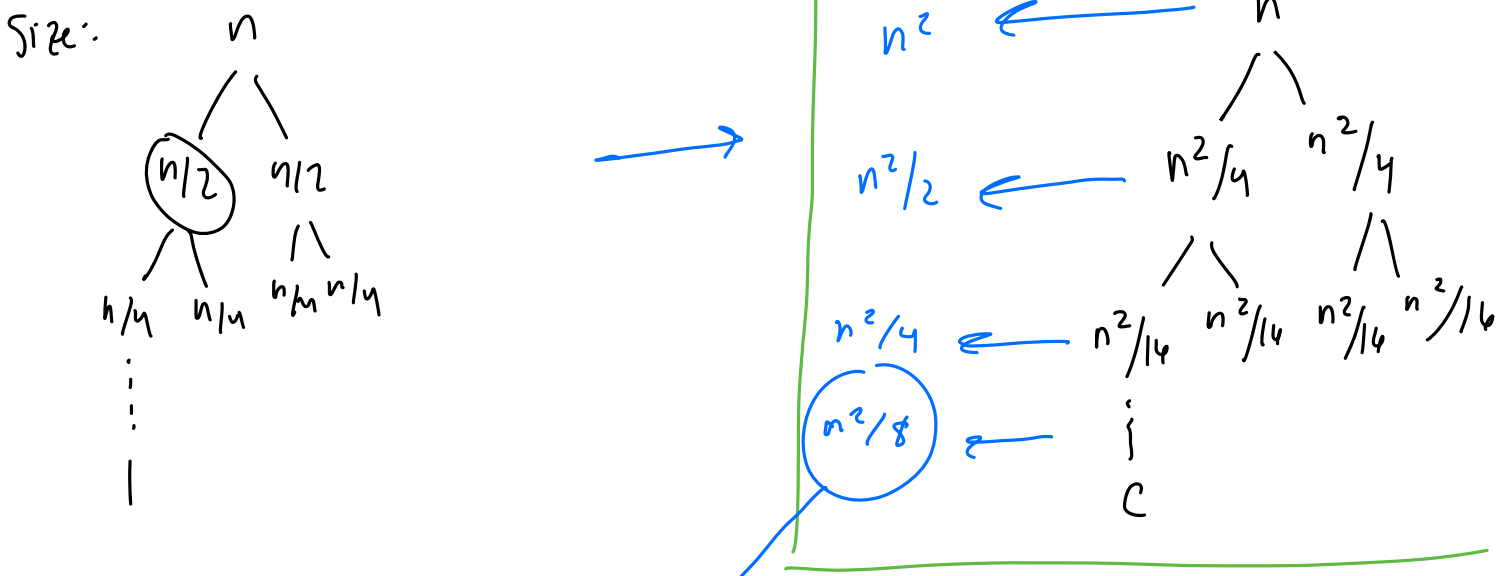
total work/sum: $n \log_2 n + cn$

alternate method to find closed form

example) $P(1) = c$

$P(n) = 2P(n/2) + n^2 \quad \forall n \geq 2$

recursion tree:



What is total work in tree?

internal: $\sum_{i=0}^{\log_2 n - 1} n^2/2^i$ (level i)

$= n^2 \sum_{i=0}^{\log_2 n - 1} \left(\frac{1}{2}\right)^i = 2n^2\left(1 - \left(\frac{1}{2}\right)^{\log_2 n}\right)$

review summations + log rules.

$= 2n^2\left(1 - \frac{1}{n}\right)$

$= 2n^2 - 2n$

leaves: $c \cdot \#\text{leaves}$

$c \cdot 2^{\log_2 n} = cn$

total work: $2n^2 - 2n + cn$

$2n^2 + n(c-2)$