The tables below are examples of what is expected for (i), (ii), (iii), and (iv) on Q 4 of HW2. We assume for simplicity that n is a power of 2.

7 Solving Recurrences

We are exploring the algorithm design technique known as **Divide and Conquer**. We'll see various algorithms that use this technique.

Running time analysis of such algorithms naturally involves *recurrences* since we may state the running time in terms of the running time on smaller inputs. Let's think more about solving recurrences to help us determine the running time of such algorithms!

7.1 Using the Recursion Tree Method (CLRS §4.4)

Example $T(n) = 2T(n/2) + cn^2$, $T(1) = c$

level	size of problem (i)	non-recursive cost of one problem (ii)		# of problems (iii)		total cost at this level (iv)
0	n	cn^2	×	1	=	cn^2
1	$n/2$	$c\left(\frac{n}{2}\right)^2$	×	2	=	$\frac{cn^2}{2}$
2	$n/4$	$c\left(\frac{n}{4}\right)^2$	×	4	=	$cn^2/4$
⋮	⋮	⋮		⋮		⋮
$d-1$	2	$c \cdot 2^2$	×	$2^{\log_2 n - 1}$ $(= n/2)$	=	$2c$
$d = \log_2(n)$	1	c	×	$2^{\log_2 n}$	=	n

Example $T(n) = 2T(n/2) + c$, $T(1) = c$

level	size of problem (i)	non-recursive cost of one problem (ii)	# of problems (iii)	total cost at this level (iv)
0	n	c	1	c
1	$n/2$	c	2	$2c$
2	$n/4$	c	4	$4c$
⋮	⋮	⋮	⋮	⋮
$d-1$	2	c	$2^{\log_2 n - 1} = \frac{n}{2}$	$\frac{n}{2} \cdot c$
$d = \log_2(n)$	1	c	n	c

Note: the number of nodes at level i in both examples below is 2^i
So at level $\log_2 n$ we have $2^{\log_2 n} = n$ leaves

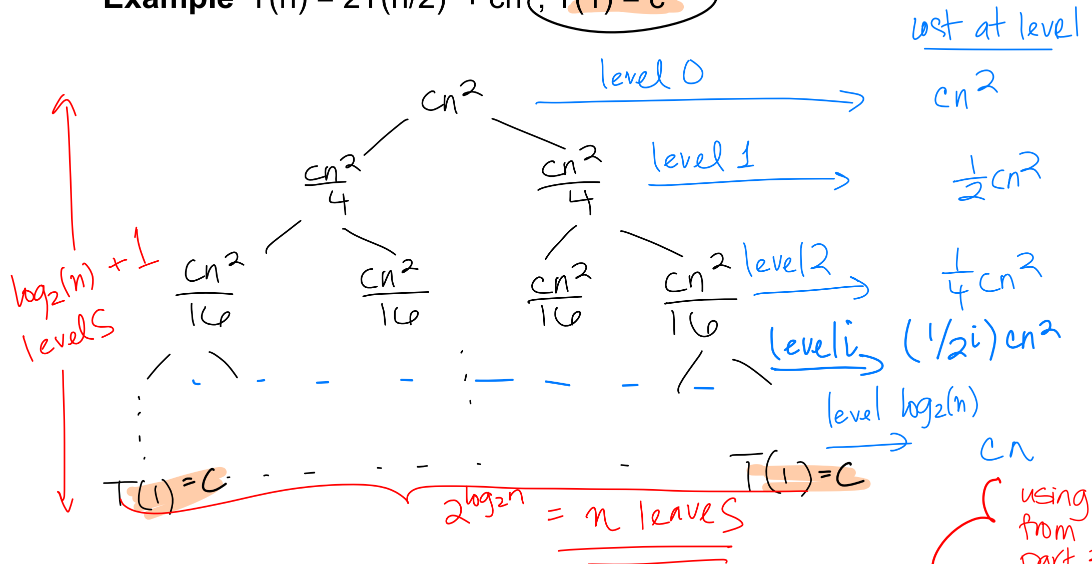
7 Solving Recurrences

We are exploring the algorithm design technique known as **Divide and Conquer**. We'll see various algorithms that use this technique.

Running time analysis of such algorithms naturally involves *recurrences* since we may state the running time in terms of the running time on smaller inputs. Let's think more about solving recurrences to help us determine the running time of such algorithms!

7.1 Using the Recursion Tree Method (CLRS §4.4)

Example T(n) = 2T(n/2) + cn^2, T(1) = c



$$\text{Total} = cn^2 + \tfrac{1}{2}cn^2 + \tfrac{1}{4}cn^2 + \dots + cn$$

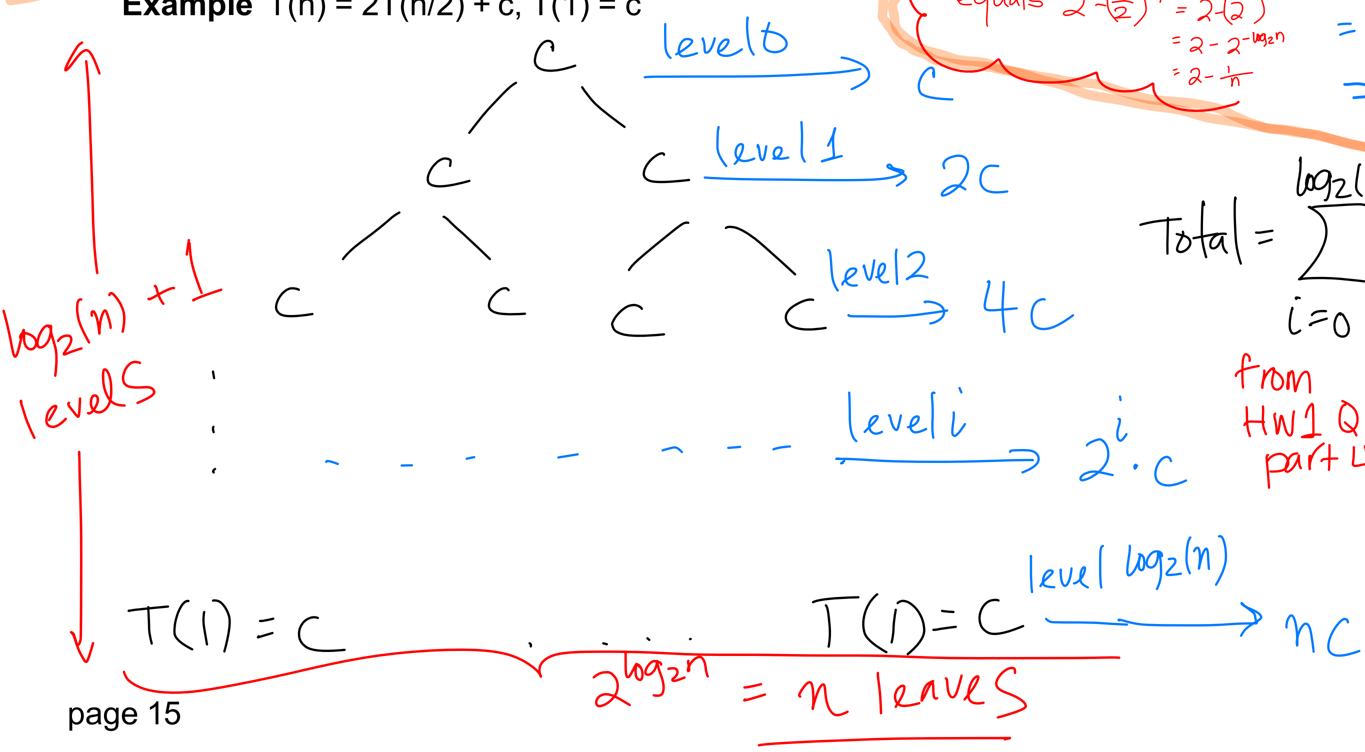
$$= \sum_{i=0}^{\log_2 n} cn^2 \left(\frac{1}{2^i}\right)$$

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

using formula from HW 1 Q 1 part 3 with $K = \log_2 n$, this equals $2 - \left(\frac{1}{2}\right)^{\log_2 n} = 2 - (2^{-1})^{\log_2 n} = 2 - 2^{-\log_2 n} = 2 - \frac{1}{n}$

$$cn^2\left(2 - \frac{1}{n}\right) = 2cn^2 + cn = \boxed{\Theta(n^2)}$$

Example T(n) = 2T(n/2) + c, T(1) = c



$$\text{Total} = \sum_{i=0}^{\log_2(n)} 2^i n = \boxed{\Theta(n)}$$

from HW1 Q1 part 4