

Solution to
Problem **3**
of handout
for Section 3

In the base case, we just return, so the work is constant. In non-base cases, we iterate over the nested loops and then make 2 recursive calls.

$$T(n) = \begin{cases} C_0 & n=0 \\ 2T(n/2) + C_2 \frac{n^2-n}{2} + C_1 & n>0 \end{cases}$$

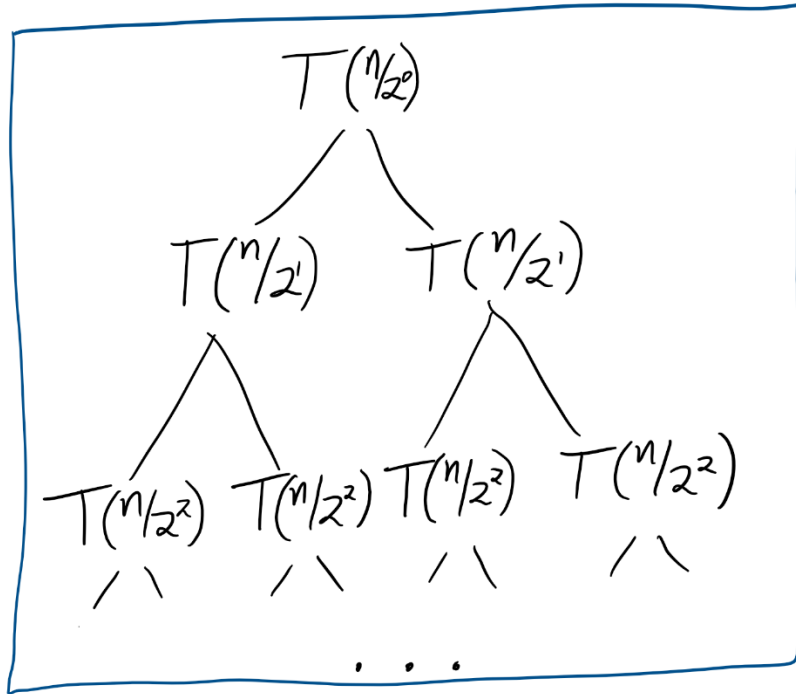
See summations sheet

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

outer loop
inner loop

"otherwise" also works as we are not expecting negative input (n usually represents size of input)

non
leaves



leaves

$T(0) \quad T(0) \quad \dots \quad T(0)$

| Level | number of nodes | work at each node |
|--------------------------|------------------------------|----------------------------------|
| 0 | $1=2^0$ | $\frac{n^2-n}{2}C_2+C_1$ |
| 1 | $2=2^1$ | $\frac{(n/2)^2-(n/2)}{2}C_2+C_1$ |
| 2 | $4=2^2$ | $\frac{(n/4)^2-(n/4)}{2}C_2+C_1$ |
| | ⋮ | ⋮ |
| $\lfloor \lg(n) \rfloor$ | $2^{\lfloor \lg(n) \rfloor}$ | C_0 |

so we get:

because level $\lg(n)$ is leaves, this adds up non-leaf work

$$\sum_{i=0}^{\lg(n)-1} 2^i \left(\frac{(\frac{n}{2^i})^2 - (\frac{n}{2^i})}{2} C_2 + C_1 \right) + C_0 \cdot 2^{\lg(n)}$$

↑ each level of the tree

↑ number of nodes at that level

↑ work at each node at that level

↑ leaf work

but we're looking for Big-O,
not exact closed form, so
we can make life a little
easier by turning $\left(\frac{\binom{n}{2^i}^2 - \binom{n}{2^i}}{2} C_2 + C_1\right)$
into $\binom{n}{2^i}^2$.

$$\sum_{i=0}^{\lg(n)-1} 2^i \binom{n}{2^i}^2 + C_0 2^{\lg(n)}$$

$$= \sum_{i=0}^{\lg(n)-1} 2^i \cdot \frac{n^2}{2^{2i}} + C_0 n$$

$$= \sum_{i=0}^{\lg(n)-1} n^2 \cdot \frac{2^i}{4^i} + C_0 n$$

$$= n^2 \sum_{i=0}^{\lg(n)-1} \left(\frac{2}{4}\right)^i + C_0 n$$

Infinite Geometric Series
is easier to deal with
than Finite Geometric Series

$$\leq n^2 \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i + C_0 n$$

we can do this because we want Big-O,
for which we need to show less than or equal to

$$= n^2 \left(\frac{1}{1 - \frac{1}{2}} \right) + C_0 n$$

$$= n^2 \left(\frac{1}{\frac{1}{2}} \right) + C_0 n$$

$$= n^2 \cdot 2 + C_0 n$$

↑ this is all we care about for Big-O,

so it's $O(n^2)$!