Study Group – Recurrences

Name: _____ Wisc id: _____

Solving recurrences

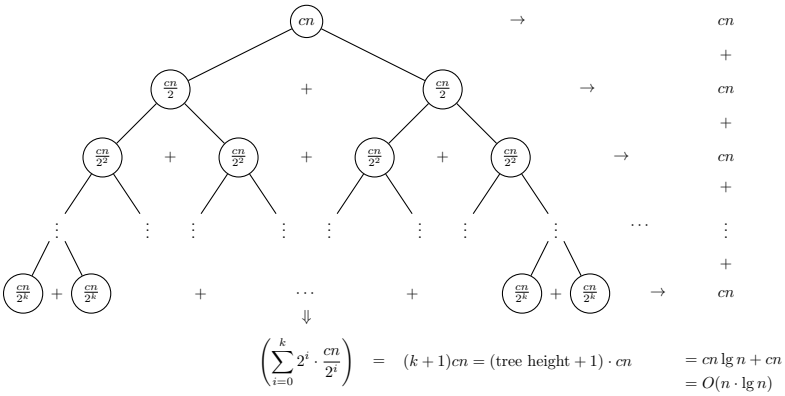
$$T(n) \leq 2 \cdot T\left(\frac{n}{2}\right) + cn; T(1) \leq c$$

Unrolling / unwinding:

$$\begin{aligned}
T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\
&\leq 2\left(2T\left(\frac{n}{4}\right) + c\frac{n}{2}\right) + cn \\
&\leq 2\left(2\left(2T\left(\frac{n}{2^3}\right) + c\frac{n}{2^2}\right) + c\frac{n}{2}\right) + cn \\
&\vdots \\
&\leq 2^k T\left(\frac{n}{2^k}\right) + kcn \\
&= nT(1) + cn\log(n), \text{ using (1)} \\
&= cn + cn\log n \\
&= O(n\log(n))
\end{aligned}$$

$$\begin{aligned}
1 &= \frac{n}{2^k} \\
\iff 2^k &= n \\
\iff k &= \log_2(n) \qquad (1)
\end{aligned}$$

Recursion Tree:



Master Theorem: If $T(n)$ is defined by a standard recurrence, of the form

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + O(n^d)$$

where $a \geq 1, b > 1, d \geq 0$.

The parameter a represents the number of recursive calls.
The parameter b represents the factor by which the input size shrinks in recursive calls.
The parameter d represents the exponent of the work done outside of the recursive calls.
Then, by the Master Theorem:

$$T(n) = \begin{cases} O(n^d \log n), & a = b^d \\ O(n^d), & a < b^d \\ O\left(n^{\log_b a}\right), & a > b^d \end{cases}$$

Problems

Find an asymptotic bound for each recurrences. Assume $T(1) = 1$ for all problems.

1. $T(n) = T(n-1) + 1$

2. $T(n) = T(n/2) + 1$

3. $T(n) = 2 \cdot T(n-1) + 1$

4. $T(n) = 4 \cdot T(n/4) + n$

5. $T(n) = 8 \cdot T(n/2) + n^2$

6. $T(n) = 6 \cdot T(n/15) + n^4$