# CSC236 winter 2020, quiz week $4_2$

**first/given name:** **last/family name:**

**utorid:**

Use the technique of unwinding to find a closed form for the following recurrence $T(n)$, assuming $n$ is a power of 2:

$$T(n) = \begin{cases} 2 & \text{if } n = 1 \\ 2n + 2T(n/2) & \text{if } n > 1 \end{cases}$$

Your closed form should be exactly equal to $T(n)$ (do not modify any of the constants in the definition). It should not recursively refer to $T$, and ideally should not involve any summations (using $\Sigma$ notation or "...").

You may use algebraic substitution and/or tree diagrams. You do not need to prove your closed form correct, but it should be reasonably clear from what you've written how you arrived at your closed form. It's recommended (but not required) to check your closed form on a small value of $n$ to verify that it works.

**Solution**

$$\begin{aligned} T(n) &= 2n + 2T(n/2) \\ &= 2n + 2(2n/2 + 2T(n/4)) \\ &= 2n + 2(2n/2 + 2(2n/4 + T(n/8))) \\ &= 2n + 4n/2 + 8n/4 + ... + 2^{\log n}(2n/2^{\log n} + T(1)) \\ &= 2n + 2n + 2n + ... + n * (2n/n + 2) \\ &= 2n + 2n + 2n + ....2n + 2n \end{aligned}$$

I observe that each term is $2n$ (including the final one). I infer that there will be $\log n + 1$ terms, giving a closed form of

$$T(n) = (\log n + 1) * 2n = 2n \log n + 2n$$

*Tree method alternative*: The tree for this $T(n)$ would be a binary tree, similar to the mergesort example in class. The root would be labelled with 2n, with 2 children representing n steps each, 4 grandchildren with n/2 steps each, etc. I observe that the total number of steps at each level is 2n, including the leaves. The total height is $\log n$, meaning the tree has $\log n + 1$ levels, giving a closed form of

$$T(n) = (logn + 1) * 2n$$