EECS 340, Breakout Session Notes, September 9, 2019

1. Begin by handing out the quiz. The students are to work alone with no notes, phones, etc. Collect the quiz at 3:40 so that you have 30 minutes to do the exercise below.

2. Ask the students to get in groups of 2 or 3 so they can work together to solve this problem: You are playing a game of "pickup sticks". There are N sticks in a pile. Each player takes turns, and there is a list of legal moves $\{m_1, \ldots, m_k\}$. For example, the first time you play the game, the legal moves could be $\{2, 3, 5, 7\}$, and the second time you play the game the legal moves could be $\{1, 3, 4\}$.

   Each player takes turns taking sticks. A player must take a number of sticks that equals a legal move. The player who takes the last stick *wins* the game. If there are too few sticks left for any legal moves (this happens when 1 is not a legal move), then the game ends in a draw.

   You are going first. There are N sticks and you have the list of $\{m_1, \ldots, m_k\}$ legal moves. Give an algorithm to determine if there is a first move that can guarantee you a win (or if not, is there one that can guarantee you at least a draw)?

3. In 5 minutes, ask if they figured out which algorithm technique to use? I suggest you ask all the groups and see if you have consensus on the algorithm or if different groups come up with different techniques. If there is not consensus, you can ask if anyone is able to show that divide and conquer does not make sense, and if anyone can prove that greedy will not (always) work.

4. Now that it is dynamic programming, the choices should be clear, but the hard part is what goes in the table. Stop in 5 minutes to ask what the recurrence should look like. Let the group share their answers. If their answer does not work, show why and let them keep working. Do not give them the recurrence right away, but see if the groups can figure it out on their own. If they do get it, write it on the board so there is a "clean" version all the groups can consult.

5. If we get to 3:55 and no group has the recurrence, give them the start. Show them that we need a $T[k]$ such that
   $$T[k] = \{W, L, D\}$$
   whether we are guaranteed to win, lose, or draw the game if we make a move with k sticks left. Have the class work out when a move is a W, a L, or a D. Here is the recurrence.
   $$T[k] = \begin{cases} W & \text{if there exists a move } m_j \text{ with } T[k - m_j] = L. \\ L & \text{if for every move } T[k - m_j] = W. \\ D & \text{otherwise} \end{cases}$$

6. The next step is the proof. If it is before 4:00, and groups have a proof, you can have them read out their proof. The key points you should be hinting at are: What are the base cases? What are the induction assumptions? And where does the induction hypothesis show up. Here is the full proof. You should start writing the proof between 4:00 and 4:05.

   Base cases: $T[0] = L$, and $T[k] = D$ for all k smaller than the smallest legal move.

   Proof: Assume $T[k - m_j]$ correctly stores whether the player making a move is guaranteed a win, loss or draw for each move m_j.

   If there is a move m_j that produces a $T[k - m_j] = L$, then by the induction hypothesis, if I take m_j sticks my opponent is guaranteed to lose, and I win. So $T[k] = W$.

   If for every move m_j with $m_j \geq k$, it results in $T[k - m_j] = W$, then by the induction hypothesis, my opponent is guaranteed to win (unless my opponent makes a mistake), and so I am guaranteed to lose. $T[k] = L$.

   If no move leads to a guaratee loss for my opponent, and not every move leads to a guaranteed win, then there must be some move m_j that has $T[m_j] = D$. By the induction hypothesis, if I take m_j sticks I am guaranteed at least a draw. So $T[k] = D$.

7. 4:08 In the last minute, ask them what the running time is? The table is size N and it takes $\Theta(k)$ to compute each value, so $\Theta(kN)$. Is this polynomial? No! It is weakly polynomial because it is exponential in the number of digits of N. As long as N does not get too big, this is polynomial. But if we allow computers to play a game with an astronomical number of sticks, then this is exponential.