CS218 HW 4 Programming

due Thursday, May 16, 23:59 PM

Problem A // ID: 259964517

Dynamic programming for minimum edit distance, but instead using min cost from the prices of each candy being added.
If the current characters in the left and right strings are equal, then dp[i][j] = dp[i - 1][j - 1]
Otherwise, the DP recurrence is:
dp[i][j] = min(dp[i - 1][j] + cost.get(str1[i - 1], 0), dp[i][j - 1] + cost.get(str2[j - 1], 0))
The algorithm is called with the candy string and its reverse to solve for the minimum cost insertions needed for the two strings to become equal.

Runtime: $O(n^2)$ due to filling out the DP array; Space Complexity: $O(n^2)$, storing the DP array.

Problem B // ID: 260048177

Minimax algorithm to find optimal play of tictactoe, given inital board state. The algorithm determines whose turn it currently is, loops through all possible placements of the mark, and recursively calls minimax on the other player's turn to determine the outcome of the board. Memoization is utilized on each board state to save exponential computations on repeat board states.

Runtime: Greatly determined by the initial board state, but there are a maximum of $3^9 = 19683$ possible states to explore given 3 options for each of the 9 squares (X, O, #); Space Complexity: 3^9 possible states to explore and memoize.