

CS218 HW 3 Challenge

due Thursday, May 9, 23:59 PM

Problem A // ID: 257359864

To determine the optimal sum of TS and TF , I check all subsets of cows, and only check the set sum against the best sum if both TS and TF are non-negative. If every best sum would include a negative smartness or fun value, return 0.

Runtime: $O(n * 2^n)$ due to the nested loop running $2^n * n = n * 2^n$ times; Space Complexity: $O(1)$, due to counters.

Problem B // ID: 257364658

Dynamic programming storing two rows at a time. Run through each possible substring from correct string (Sheldon), to modified string (Raj). Check through each of the modifications: 1) remove, 2) add, 3) copy a letter from the correct string. If the characters at indices don't match, then the copy incurs a +1 to the mistakes, otherwise just the same mistakes as $[i - 1][j - 1]$.

Runtime: $O(n^2)$ due to nested for loop fill out dynamic programming array; Space Complexity: $O(n)$, due to only $2n$ rows of the DP array being stored at any time.

Problem C // ID: 257368937

I recursively check all possibilities of substrings being reversed or not. The lexicographically small string (of the leaves of the tree) will be the output.

Runtime: Given n as the length of the string, $O(2^n)$ due to recursively checking either action on all n substrings; Space Complexity: $O(n)$, for n levels of recursion.

Problem D // ID: 257387272

Use DFS to pop nodes containing seen courses and the corresponding credit total. Loop through all courses for each node, only add a candidate course if it isn't already seen and the pre-requisite is met.

Runtime: $O(2^n)$ due to testing all 2^n course selection options; Space Complexity: $O(2^n)$, for storing all 2^n nodes.