Palindrome

Based on the examples given the palindrome solution is going to follow longest common sequence rules where we are comparing two strings and seeing if a palindrome can be made from them and output how many letters need to be removed inorder to have a 1 string palindrome. A brute force solution would have a time complixity of O(n2^m) because we are doing n work for each letter. In example two it suggests that when the x = c is met with y = e , y iterates until they reach x == y at c this 2^m .

(Attempted Extra Credit I - 7 and I -8)

A second better approach would be to use Dynamic Programming and use a bottom-up approach and solve as if we were solving a longest common subsequence by using a matrix to compare the forward and backwards version once the LCS is found then use the results and calculate for k. The resulting time would fall in O(m*n) which is an improvement from the Brute Force Approach.

Pattern Match

I'll be using a Brute Force Approach. My approach will loop until finished running at a time complexity of O(n). There are some restrictions as it iterates through. If the previous is '*' then check {j – 1 , j , j+1 … j + n}, '?' must equal a character, when a false is met return false. Otherwise the pattern and string are valid return true.

Get Tesla

We could apply methods that we learned from Greedy Algorithms to solve this problem. Since our movement is restricted to either down or right we don't need to worry about any cycles and can focus on soley taking the best option given two options. The restrictions here being starting tile and final tile impact the problem. To find a solution we have the algorithm preform greedy choices to reach the end we should be left with an total. For example in the example matrix the path taken is -1 ->10->-5->-2->-3 = -1, the answer then would be 2. To address the second constraint the result is then compared with the starting tile and for example instead of -1 it was a -2 then the program compares 2 + (-2) = 0 or 2 + (-5) = -3 then the program takes the results here and takes the absoult value and + 1 and then adds it to the previous result of 2 and outputs total hp needed to start to win.

------------------- Debrief ------------

I spent 4.5 hours on the assignment

Medium

85%

It forces us to think critically about the content we've learned in class and try and apply. We're not limited by subject and instead have to think if there is another way to solve with out relying on a brute force approach.