

CSCA48 Exercise 8

Due: March 23, 2018. 5:00pm

More recursion examples... YAY!!!

In a file called `ex8.py`, complete the following functions. All functions must be **recursive**, and only the **Easy** version of each function will be marked. (The **Hard** versions are just for fun).

`edit_distance(s1, s2):`

The edit distance of two strings is defined as: **The minimum number of single character changes that would be needed to turn `s1` into `s2`.**

Easy: assume that `s1` and `s2` are the same length, and the only character changes available are replacing one character with another.

Hard: assume `s1` and `s2` may be different lengths, and we can also add and delete characters.

`subsequence(s1, s2):`

`s1` is a subsequence of `s2` iff `s2` can be made equal to `s1` by removing 0 or more of its characters. **Example:** `subsequence('dog', 'XYZdABCo123g!!!')` should return `True`.

Easy: Return `True` iff `s1` is a subsequence of `s2`.

Hard: Return a list of integers representing the characters of `s2` that must be deleted to turn it into `s1`.

`perms(s):`

Easy: Given a string `s`, return a set of all possible permutations of the letters in `s`.

Hard: Assuming that `s` is a string of 0s and 1s, return a list of the permutations ordered such that they form a Gray Code (a sequence of binary numbers such that each successive pair of numbers differ by exactly 1 bit).

Submission:

Submit your file containing all three functions to MarkUs. Remember that **Only the Easy versions will be marked**. If you wish to submit the **Harder** versions, ensure that they have different function names (or else the auto-marker will mark them as incorrect).