Challenge 2: Longest Common Substring

In this lesson, you will work on an interesting dynamic programming problem, the longest common substring.

We'll cover the following

Problem statement

Given two strings, we want to find the length of the longest substring common in both these strings. For example, if we have two strings, `"hello elf"` and `"hello yourself"`, we can see two prominent substrings: `"hello "` and `"elf"`. Since `"hello "` is longer, this will be the longest common substring for the given pair of strings.

Input

Your program will take two strings, `str1` and `str2`, each of length greater than zero as input.

```
str1 = "hello elf"
str2 = "hello yourself"
```

Output

Your program should return the length of the longest common substring.

```
lcs("hello elf", "hello yourself") = 6
lcs("hel", "elf") = 2
```

Coding challenge

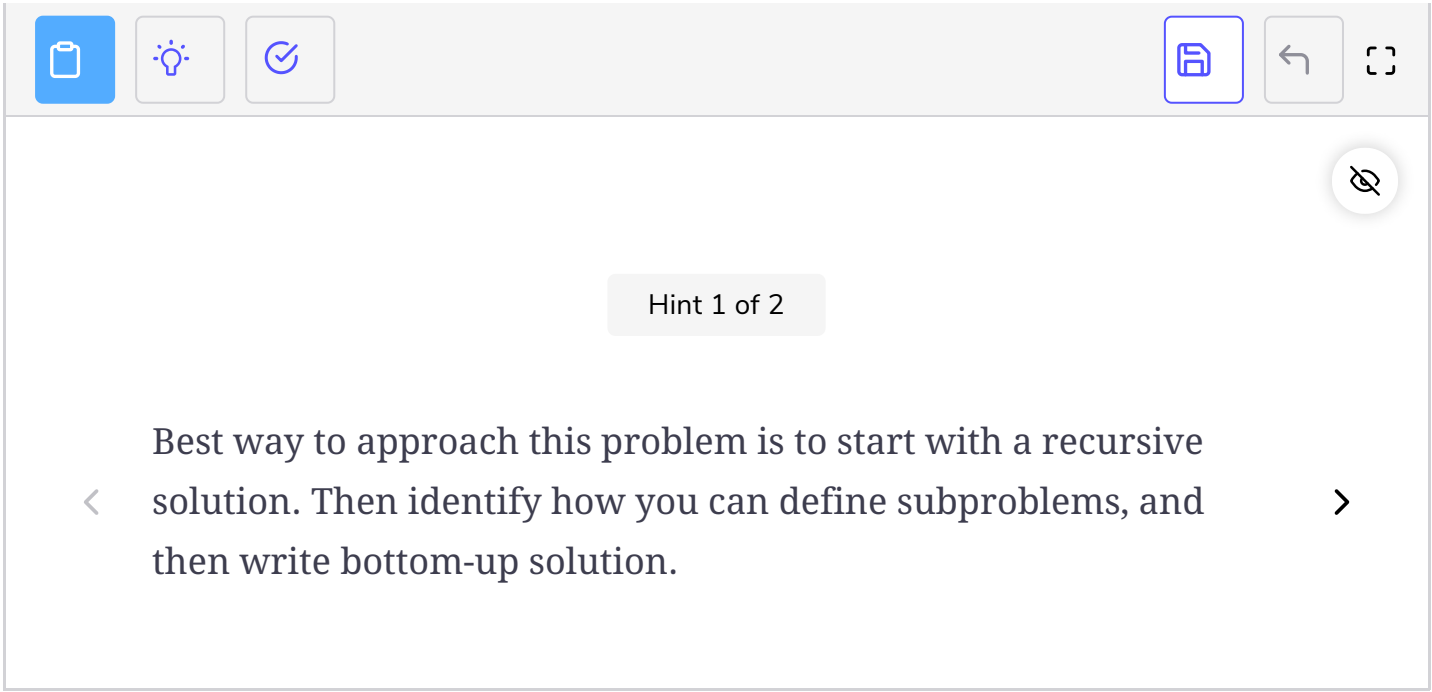
Think about some basic examples to start with and then write a simple recursive algorithm. Slowly build up to the bottom-up solution.

You may check your recursive algorithm by setting `stressTesting` to `False`. Similarly to check only top-down implementation, set `testForBottomUp` to `False`. Best of luck!

```
def lcs(str1, str2):
  # write your code here

  return -1

stressTesting = True  # to only check if your recursive solution is correct, set it to false
testForBottomUp = True   # to test a top down implementation set it to false
```



In the next lesson, we will review the solution to this coding challenge.