Sreeya Gambhirrao

AP Computer Science A

February 27, 2022

Challenge Program

What is Recursion?

Recursion is the process of repeating a job within the job itself. Recursions require you to look at the big picture and work your way down into smaller parts and details. Recursions are used a lot in mathematics. Factorials are examples of recursion.

What is Divide and Conquer?

To best complete recursion projects'; and any recursive method project in general, you can use the Divide and Conquer method. The divide and conquer method requires you to **Divide** the problem into smaller subproblems. **Conquer** those sub-problems to find a solution. Finally, you can put the pieces together to create your masterpiece. This method helps you complete your recursive project in an easier way.

Divide and Conquer Steps:

Divide this project into smaller problems of the same project.

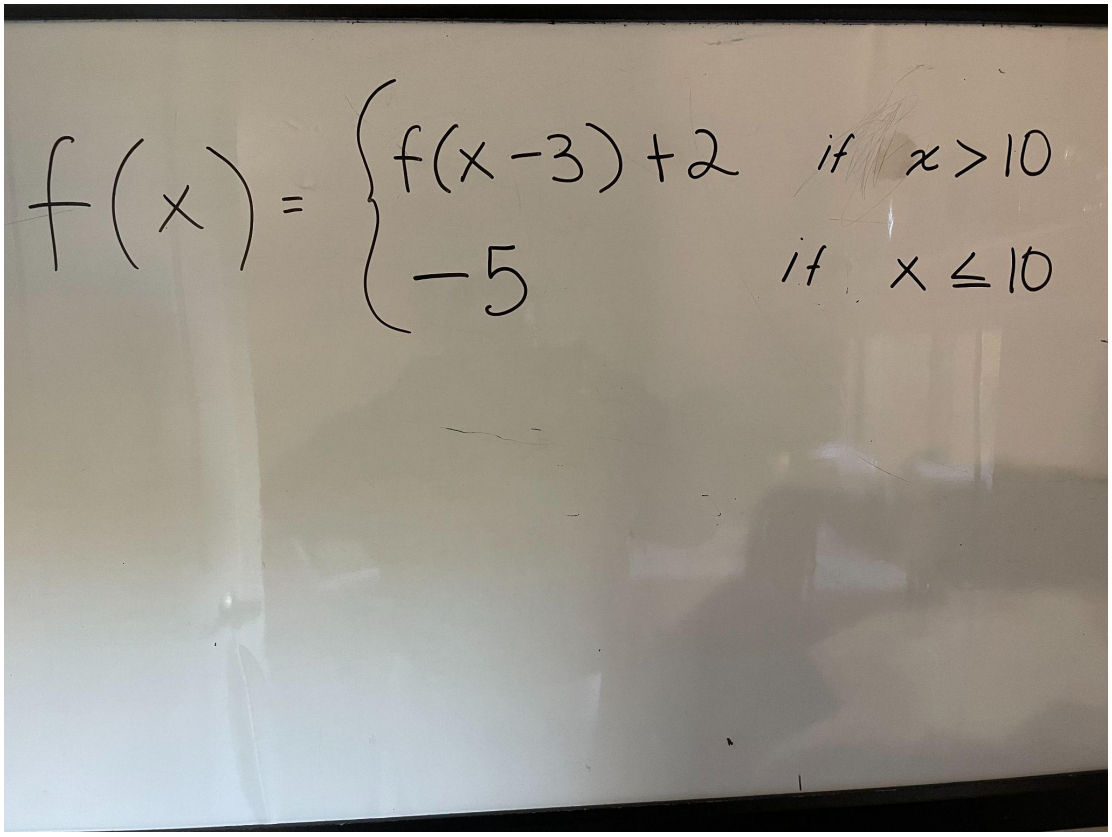
Find the solutions to each subproblem.

Write pseudocode for the recursive method. (Make sure to write the base case)

Write pseudocode for the main method

You are ready to write your code once you have confirmed pseudocode!

A great example of recursion is piecewise functions. You may have already learned about piecewise functions in math. If not, a piecewise function is a function that is defined by smaller subfunctions. Take a look at the following example below.



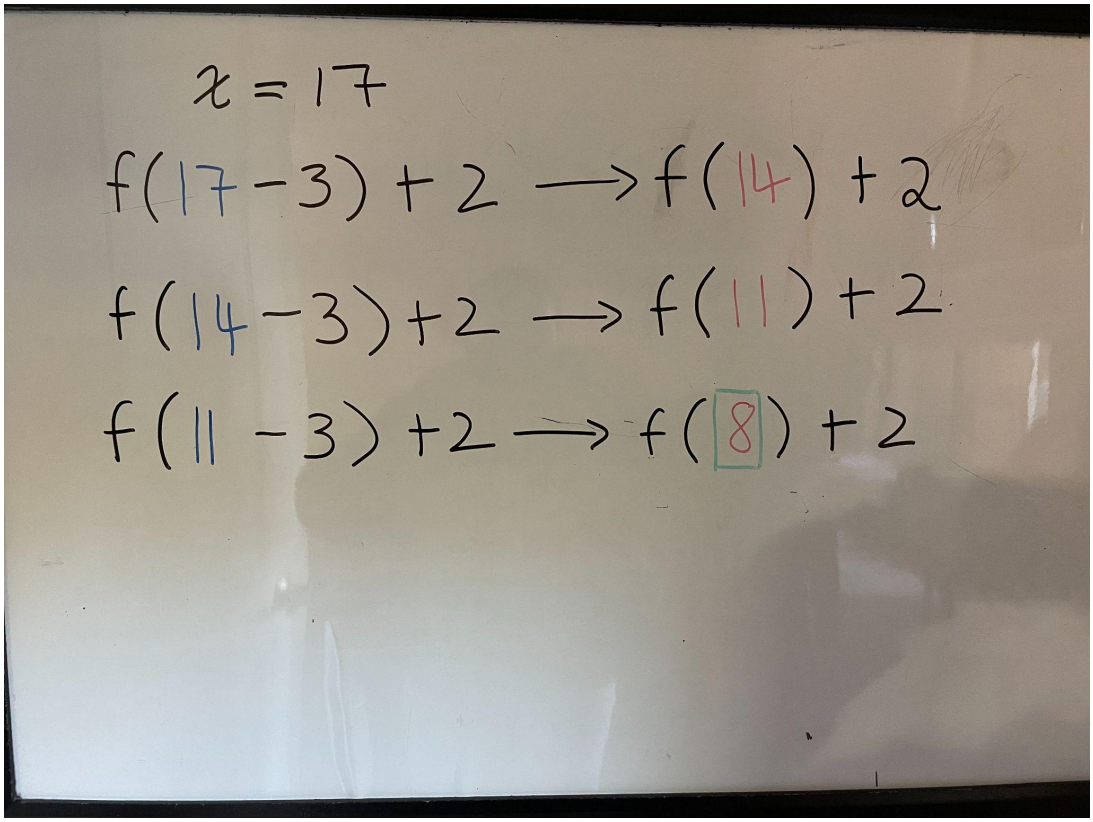
You can solve this problem using the S-S-S Strategy.

What is the S-S-S Strategy?

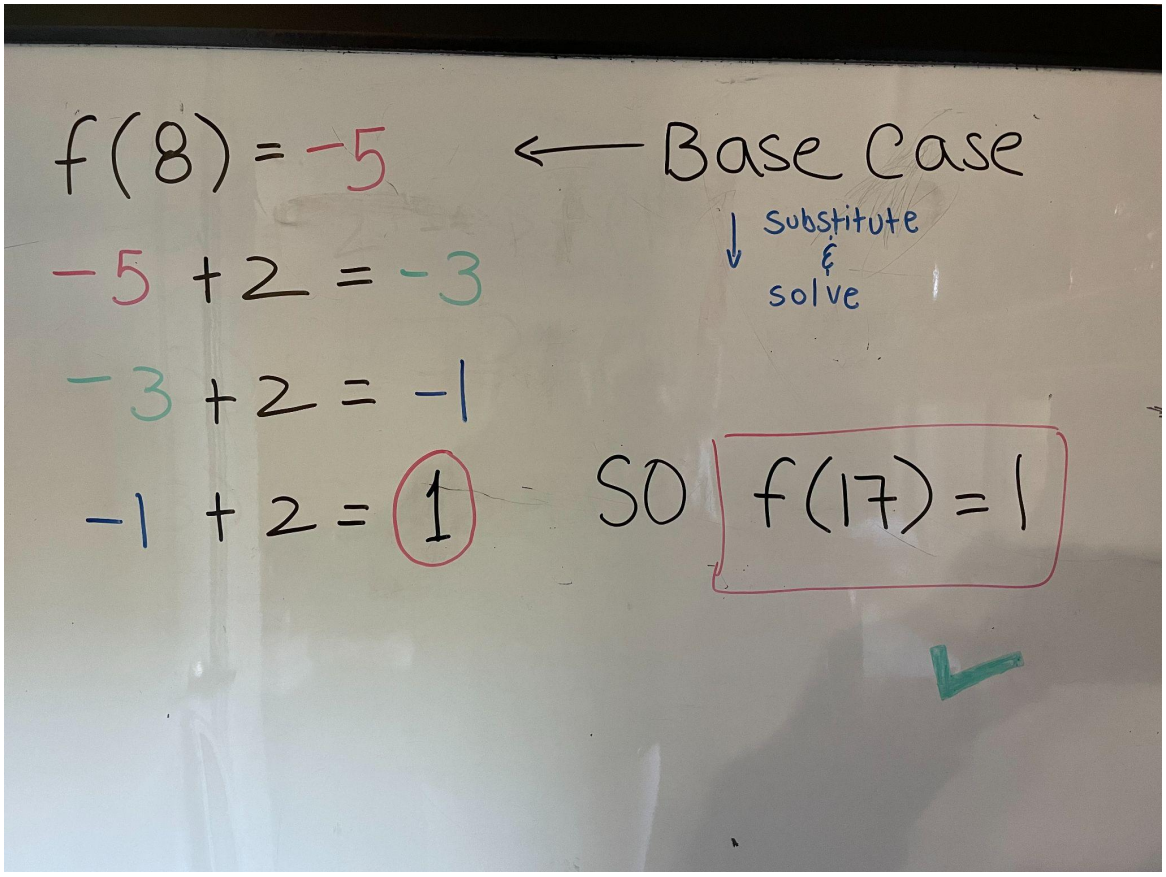
The S-S-S Strategy stands for **Simplify**, **Substitute**, and **Solve.** First, **simplify** the expression through a series of calculations until you have reached the base case. Then, **substitute** the base case into the last unsolved equation. Lastly, **solve** the calculations as you work your way up until you find your solution.

Solution:

Lets solve the above equation when x = 17.



8 is less than or equal to 10 so this is our base case. Now we have to substitute it and work our way back up by solving. See below for calculations.



So our solution is f(17) = 1.

Assessment Instructions

Purpose: This project takes a user inputted DNA sequence and uses recursion to find its complement DNA strand.

Instructions: Write a recursive program to determine the complement strand of a DNA strand.

1. Create a project in BlueJ called ComplamentDNA in the Unit02 Assessments folder.
2. Create a class called RecursiveDNA and a class called RecursiveDNATester in the newly created project
3. Prompt the user to enter a DNA sequence strand such as ACGTGACC…
4. Allow the user to continue entering their sequence until they choose to quit

5. Write a recursive method to determine the complement strand of their input sequence
6. Print the complement strand

Grading: The assessment will be graded by the following rubric.

Grading Rubric	**Pts**
Comments include name, date, and purpose of the program	1
Recursive method header written correctly	4
Base case written correctly	3
Recursive call written correctly	5
Allows the user to input sequence	1
Continuously prompts the user to enter input	2
Main() method in tester class correctly invokes recursive method	1
Constructor correctly written	1
Output is correct	1
No compiler or runtime errors	1
Thoughtful PMR included	1
Total	21

Solution(in pseudocode):

RecursiveDNA class

If character = A, then return T.

If character = T, then return A.

If character = C, then return G.

If character = G, then return C.

If character is not any one of these letters, then return some type error for the user.

RecursiveDNATester

In a while loop(user chose to continue):

- Use scanner.in to ask the user to enter input
- Create a new object of type RecursiveDNA.
- Invoke the recursive method with the object
- Create an if loop for if the character is A, T, C, or G
- Add the characters to a string
- Print the complement sequence
- Ask the user if they want to continue or quit)

If they choose quit:

Outside of while loop:

- Print a message that indicates the user that they have quitted.