Problem 1. “Another Flipping Algorithm”

Imagine that you are given a stack of N pancakes each a different size, and your task is to arrange the stack in order such that every pancake rests on a larger one, with the largest on the bottom. However, the only operation that you are allowed to rearrange the stack is to insert a spatula under one pancake and flip over the entire stack of pancakes above it.

(A) Describe an algorithm to accomplish the desired task using only the allowed operation for any given pancake stack (explain where to place the spatula and perform the flip for each step).

(B) Consider the following 4-pancake stack. Give the result of your algorithm after each flip.

(C) How many steps, as a function of the number of pancakes, N, does your algorithm require for the worst-case N-pancake stack?

(D) Draw a diagram of the 5-pancake stack that requires the maximum number of flips using your algorithm.

(E) Consider the following series of flips for the pancake stack shown in part B: first flip the top 3, then the top 2, then the whole stack. Did your algorithm require fewer steps, more steps, or the same number of steps?

By the way, the first technical paper authored by Bill Gates, the founder of Microsoft (a.k.a. the richest man in the world) concerned placing a bounds on the number of flips required to solve this problem for worst-case configuration of N pancakes.

Problem 2. “Ordering up a Good Algorithm”

Suppose you are given four versions of an algorithm for analyzing N objects. Algorithm A takes 64 N steps, Algorithm B takes 4 N^2 steps, Algorithm C takes 16 N log_2(N), and Algorithm D takes 2^N steps.

(A) What is the smallest value of N where Algorithm A takes fewer steps than Algorithm B? What is the smallest value of N where Algorithm A takes fewer steps than Algorithm C? What is the smallest value of N where Algorithm A takes fewer steps than Algorithm D?

(B) For large values of N which algorithm is faster (takes fewest steps) B or C? C or D?

(C) Suppose that a new algorithm is discovered that takes (2 N + 2)(2 N - 1) steps. To which of the 4 given algorithms is its performance closest to for large values of N?
Problem 3. “Python Twists”
What do the following Python statements generate as outputs?
(A)   for i in range(10):
       print i, 2**i

(B)   “Hello”[1:3] + “Rosey”[-3:-1]

(C)   def fact(n):
       for i in range(n-1,1,-1):
          n = n * i
       return n

       fact(7)

(D)   def rev(a)
       if len(a) < 2:
          return a
       mid = len(a)/2
       return rev(a[mid:]) + rev(a[:mid])

       rev(“This is a test”)
       rev(range(8))

Project #3: “Leave the Last Cookie!”
The “Cookie Game” is played as follows. A batch of cookies is baked and placed on a plate.
Players alternate turns taking 1, 2, or 3 cookies from the plate. The object of the game is to leave
the last cookie, for whoever takes it must bake the next batch.

In this project, you will write a computer version of the “Cookie Game” where a single human
player and the computer alternate turns. The computer begins by asking how many cookies are in
the current batch. It then makes it’s choice of how many cookies to take and tells the human
player how many are left. Next, the computer asks the human player how many cookies she wants
to take, it then verifies that her input is valid, updates the number of cookies, and assesses
whether the game has been won. If more than one cookie remains, the computer will make its
next choice, and then allows the human player to go again. This process continues until the game
is eventually won.

It is most important for your program to be correct. For example it should not allow the player to
take more cookies than are allowed, or more than are available. To maximize you grade on this
project you should try to make your computer player’s strategy as good as possible.