# ECS 32B: Fall 2018
# Homework Assignment 3

**Due Date:** No later than Friday, November 30, 9:00pm. Submit your solutions via Canvas in one file named "hw3.py". Expect that you will be given Homework Assignment 4 before Homework Assignment 3 is due.

**1.** A number of different sized pancakes are stacked. A sorted pancake stack is defined as having the smallest pancake on top, the second smallest pancake under the smallest pancake, etc. The only tool provided is a spatula that will flip any top partition of the stack, including the entire stack. Note that there only ever exists *one* stack of pancakes, not multiple smaller, independent stacks. Larger pancakes may be on top of smaller pancakes and vice versa.

a) If every flip took one unit of time to complete, exactly how many flips or units of time are required in the worst-case (i.e., for the worst-case arrangement of pancakes) to sort the stack? Aim for a good algorithm. Express your answer, $T(n)$, as a function of the number of pancakes. Make your answer easy to find in the comments for your solution to part b, below.

b) Assume that an arbitrary stack of n pancakes is represented as a Python list of numbers, with each number representing the size of the pancake. When sorted, the smallest pancake (with the smallest number) will be at index 0 in the list, and the largest pancake will be at index n – 1. Write a Python function named `pancakeSort` that expects one argument, a possibly unsorted list of numbers representing pancakes. that uses the flipping technique described above to sort the list.

**2.** To gain familiarity with bubble sort, which we did not talk about in class, do programming exercise 10 from chapter 5: Implement the bubble sort using simultaneous assignment. Name your function `bubbleSort`.

**3.** And for a little bit of practice with merge sort, do programming exercise 14 from chapter 5: Implement the `mergeSort` function without using the slice operator.