Homework 4

- **Submit your solutions electronically on the course Gradescope site as PDF files.** If you plan to typeset your solutions, please use the LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera). We will mark difficult to read solutions as incorrect and move on.
- **Every homework problem must be done *individually*.** Each problem needs to be submitted to Gradescope before 6AM of the due data which can be found on the course website: https://ecealgo.com/homeworks.html.
- For nearly every problem, **we have covered all the requisite knowledge required to complete a homework assignment prior to the "assigned" date.** This means that there is no reason not to begin a homework assignment as soon as it is assigned. Starting a problem the night before it is due a recipe for failure.

Policies to keep in mind

- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use, and you ***must*** write everything yourself in your own words. See the academic integrity policies on the course web site for more details.
- **Being able to clearly and concisely explain your solution is a part of the grade you will receive.** Before submitting a solution ask yourself, if you were reading the solution without having seen it before, would you be able to understand it within two minutes? If not, you need to edit. Images and flow-charts are very useful for concisely explain difficult concepts.

See the course web site (https://ecealgo.com) for more information.

If you have any questions about these policies,
please don't hesitate to ask in class, in office hours, or on Piazza.

1. Solve the following recurrence relations. For parts (a) and (b), give an exact solution. For parts (c) and (d), give an asymptotic one. In both cases, justify your solution.

   (a) $W(n) = W(n-1) + 2\log n + 1; W(0) = 0$

   (b) $X(n) = 5X(n-1) + 3; X(1) = 3$

   (c) $Y(n) = Y(n/2) + 2Y(n/3) + 3Y(n/4) + n^2$

   (d) $Z(n) = Z(n/15) + Z(n/10) + 2Z(n/6) + \sqrt{n}$

2. Suppose you are given a stack of n pancakes of different sizes. You want to sort the pancakes so that smaller pancakes are on top of larger pancakes. The only operation you can perform is *flip* - insert a spatula under the top k pancakes, for some integer k between 1 and n, and flip them all over.

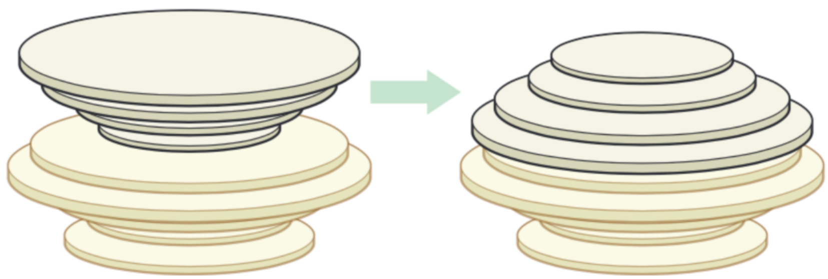

Figure 1.19. Flipping the top four pancakes.

   (a) Describe an algorithm to sort an arbitrary stack of n pancakes using $O(n)$ flips. *Exactly* how many flips does your algorithm perform in the worst case? [*Hint: This problem has nothing to do with the Tower of Hanoi.*]

   (b) For every positive integer n, describe a stack of n pancakes that requires $\Omega(n)$ flips to sort.

   (c) Now suppose one side of each pancake is burned. Describe an algorithm to sort an arbitrary stack of n pancakes, so that the burned side of every pancake is facing down, using $O(n)$ flips. *Exactly* how many flips does your algorithm perform in the worst case?

3. Suppose we are given an array $A[1..n]$ of n integers, which could be positive, negative, or zero, sorted in increasing order so that $A[1] \le A[2] \le \cdots \le A[n]$. Suppose we wanted to count the number of times some integer value x occurs in A. Describe an algorithm (as fast as possible) which returns the number of elements containing value x.

4. Given an arbitrary array $A[1..n]$, describe an algorithm to determine in $O(n)$ time whether A contains more than $n/4$ copies of any value. **Do not use hashing, or radix sort, or any other method that depends on the precise input values.**