CSC236 Summer 2017
Assignment #2: Analysis of Recursive Algorithms
Due July 13th, by 6:00 pm

The aim of this assignment is to give you some practice analyzing runtime and proving correctness of recursive algorithms.

For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks will be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions. You should clearly cite any sources or people you consult, other than the course notes, lecture materials, and tutorial exercises.

Your assignment must be typed to produce a PDF document **A2.pdf** (hand-written submissions are not acceptable). You may work on the assignment in groups of 1, 2, or 3, and submit a single assignment for the entire group on MarkUs.

1. Consider the following recurrence relations:

   (a)
   $$S(n) = \begin{cases} 1, & \text{if } n = 0 \\ (S(n-1))^2 + 2S(n-1) & \text{if } n > 0 \end{cases}$$

   Hint: let $T(n) = S(n) + 1$, and try to find a closed form of $T(n)$ first.

   (b)
   $$S(n) = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ S(n-2) + 2n - 1 & \text{if } n > 1 \text{ and } n \text{ is even} \\ S(n-2) + 3n & \text{if } n > 1 \text{ and } n \text{ is odd} \end{cases}$$

   For each recurrence relation, find a closed from for $S(n)$ and briefly describe how you get the closed form. Then prove the closed form is correct. Based on the closed form, state a conjecture for $f(n)$ such that $S(n) \in \Theta(f(n))$, and prove the bound is correct.

2. Assume $S(n)$ is monotonic non-decreasing. We do not know the value of $S(n)$ for every $n \in \mathbb{N}$ except when $n = 2^k$ for some $k \in \mathbb{N}$, in which case $S(n) = n \log n + 3n - 5$. Show that $S(n) \in \Theta(n \log n)$ for all $n > 0 \in \mathbb{N}$, not just special cases.
   Hints: you may need to use one of the following facts (or both, or none of them) in your proof:

   (a) $\forall n > 1 \in \mathbb{N}, \exists k \in \mathbb{N}$, such that $2^{k-1} \leq n \leq 2^k$.

   (b) for functions $f(n), g(n) : \mathbb{N} \to \mathbb{R}^+$, if $\lim_{n\to\infty} f(n)/g(n) = 1$, then $f(n) \in \Theta(g(n))$

3. Consider the problem of finding the maximum sum of consecutive values in a sequence of integers. For example, the maximum consecutive sum in $2, -5, 3, -1, -1, 4$ is $3 + (-1) + (-1) + 4 = 5$, and the maximum consecutive sum in $1, -2, 3$ is 3. Let n be the number of integers in the sequence.

   (a) Give a brute force algorithm that solves the problem with worst-case running time $O(n^2)$, and informally state why your algorithm would have worst-case running time $O(n^2)$.

   Consider a Divide & Conquer approach to this problem, based on the idea that, for any sequence, either the middle element is included in the maximum consecutive sum, or it is not. Based on the idea above, consider how many subproblems the original problem would be split into, and approximately how large each subproblem would be.

(b) Give a recurrence $T(n)$ that represents the worst-case running time for an algorithm based on this idea. If the Master Theorem was applied to this algorithm, what would a and b be for $T(n)$?

(c) Give a D&C algorithm for this problem (i.e. write the pseudo-code), with worst-case runtime $T(n) \in \Theta(n \log(n))$.

(d) Prove that $T(n) \in \Theta(n \log(n))$.

Note: The optimal solution to this problem, which is not a D & C algorithm, should cost time $\Theta(n)$ if the maximum sum is calculated for sublists with constant length.

4. (a) Let $T(n)$ denote the number of distinct ways that a postage of n cents, where $n \geq 4$ and n is even, can be made by 4-cent and 6-cent stamps. For example, if $n = 12$, then we can use 3 4-cent stamps or 2 6-cent stamps, and there are no other possible ways. So $T(12) = 2$. Give a recursive algorithm (i.e. write the pseudo-code) to compute $T(n)$ for $n \geq 4$ and n is even. Briefly explain why the algorithm is correct but no formal proof is required.

   (b) Now assume we have 10-cent stamps in addition to the previous 2 kinds. Find a recurrence relation, $S(n)$, for the number of distinct ways that a postage of n cents, where $n \geq 4$ and n is even, can be made by 4-cent, 6-cent and 10-cent stamps.

   (c) Without using a closed form, prove $S(n)$ in part (b) is non-decreasing.

5. Proof of Correctness for iterative algorithms.

   (a) Design an iterative closest pair algorithm for finding the closest pair of points in 2D.
   Precondition: Input is a list of $n \geq 2$ points in the form $(x_i; y_i)$, where $x_i, y_i \in \mathbb{R}$
   Postcondition: Return a closest pair of points. If more than one pair of points with the same smallest distance, return any pair.
   Note: full mark will be given to only algorithms with complexity $O(n^2)$ or better.

   (b) Find complexity class of your algorithm in part (a) and briefly explain why.

   (c) Prove the correctness of your algorithm. You should first define a loop invariant, and then prove the partial correctness. Finally show that the algorithm will terminate.