CS 584/684 Algorithm Design and Analysis

Homework 3

Portland State U, Winter 2021 — *01/19/21*

Lecturer: Fang Song — *Due: 01/26/21*

Instructions. This problem set contains 4 pages (including this cover page) and 3 questions. A random subset of problems will be graded.

- Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct, and you will get significantly more partial credit if you clearly identify the gap(s) in your solution. It is good practice to start any long solution with an informal (but accurate) summary that describes the main idea. You may opt for the "I take 15%" option.
- You need to submit a PDF file before the deadline. Either a clear scan of you handwriting or a typeset document is accepted. You will get 5 bonus points for typing in LaTeX (Download and use the accompany TeX file).
- You may collaborate with others on this problem set. However, you must **write up your own solutions** and **list your collaborators and any external sources** for each problem. Be ready to explain your solutions orally to a course staff if asked.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

1. (Sumerians' multiplication algorithm) The clay tablets discovered in Sumer led some scholars to conjecture that ancient Sumerians performed multiplication by reduction to *squaring*, using an identity like

$$x \cdot y = (x^2 + y^2 - (x - y)^2)/2\,.$$

In this problem, we will investigate how to actually square large numbers.

   (a) (10 points) Describe a variant of Karatsuba's algorithm that squares any n-digit number in $O(n^{\log 3})$ time, by reducing to squaring three $\lceil n/2 \rceil$-digit numbers. (Karatsuba actually did this in 1960.)

   (b) (10 points) Describe a recursive algorithm that squares any n-digit number in $O(n^{\log_3 6})$ time, by reducing to squaring six $\lceil n/3 \rceil$-digit numbers.

   (c) (10 points (bonus)) Describe a recursive algorithm that squares any n-digit number in $O(n^{log_3 5})$ time, by reducing to squaring only five $(n/3 + O(1))$-digit numbers. [Hint: What is $(a + b + c)^2 + (a - b + c)^2$?]

2. (10 points) (Cycles) Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one (not all cycles, just one of them). The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.

3. An *Euler tour* of a strongly connected, directed graph $G = (V, E)$ is a cycle that traverses each *edge* of G exactly once, although it may visit a vertex more than once.

   (a) (10 points) Show that G has an Euler tour if and only if in-degree$(v) =$ out-degree(v) for each vertex $v \in V$.

   (b) (10 points) Describe an $O(|E|)$-time algorithm to find an Euler tour of G if one exists.