PAWS 2025: ANALYSIS AND IMPLEMENTATION OF ALGORITHMS IN NUMBER THEORY
PROBLEM SET 1

THOMAS BOUCHET, KATE FINNERTY, ASIMINA S. HAMAKIOTES, YONGYUAN HUANG

The goal for Problem Set 1 is to get more comfortable with $O(n)$ notation and practice systematically thinking about algorithms. The questions are loosely in ascending order of difficulty. Feel free to skip around and try whatever exercises would be the most helpful for you. Try as many as you can but don't feel like you need to complete them all!

1. Beginner Problems

Question 1: Prove Lemma 1.2 from the lecture notes:

Lemma. *Let $f(n)$, $g(n)$, $a(n)$, and $b(n)$ be functions satisfying $f(n) = O(g(n))$ and $a(n) = O(b(n))$. Then*

$$f(n) + a(n) = O\big(\max(|g(n)|, |b(n)|)\big)$$

and

$$f(n)a(n) = O\big(g(n)b(n)\big).$$

Question 2: (Lecture 1, Exercise 23)

Algorithm 1 Euclidean Algorithm for gcd

Given integers $a \geq b > 0$, compute $g := \gcd(a, b)$.

1. Set $r_0 := a$ and $r_1 := b$.
2. Set $i := 1$ and while $r_i \neq 0$, do the following
   (a) Set $r_{i+1} := \text{rem}(r_{i-1}, r_i)$ and $i := i + 1$.

Return r_{i-1}.

Explain why this algorithm terminates and correctly computes the greatest common divisor.

Question 3: (Lecture 1, Exercise 20) Write and analyze an algorithm that implements naive multiplication for integers. What is the complexity of your algorithm, in terms of the number of bits m and n of the integers you are multiplying?

Question 4: Let $M = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix}$. Using Gaussian elimination-like algorithms, can you compute (by hand):

(a) all possible solutions to $MX = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$,
(b) its determinant,
(c) its inverse if it exists.

Check your results with Magma!

2. Intermediate Problems

Question 5: (Lecture 1, Exercise 21) Describe the classical long division algorithm for binary integers. Prove that the complexity of your algorithm is $O((n-m)m)$, where m and n are the number of bits of the integers.

Question 6: Use Questions 3 and 5 to prove that the Euclidean Algorithm (Algorithm 1) for positive integers a, b with n, m bits, respectively, has complexity $O(mn)$.

Question 7: Compute a function $f(n)$ such that $O(f(n))$ represents the number of field multiplications needed to compute the product of two square $n \times n$ matrices. Why do we not care about the number of addition operations?

Question 8: Implement Gaussian elimination & inverse for square matrices in Magma. Compare the speed of your implementation with Magma built-in intrinsics on big inputs (big matrices and/or big entries).

3. Advanced Problem

A nice way to compute the complexity of divide-and-conquer algorithms is to use the Master theorem, of which we present a simpler version, tailored to our needs.

Theorem 1. *We suppose given a recurrence relation of the form*

$$T(n) = aT(n/b) + O(n^c),$$

where $a \geq 1$, $b > 1$, and $c < \log_a(b)$. Then we have

$$T(n) = O(n^{\log_a(b)}).$$

Question 9: In this problem, we study Karatsuba's algorithm for the multiplication of integers, which relies on a divide-and-conquer approach.

Let x and y be two integers with even length $n \in \mathbb{Z}_{>}0$ in a base $B \in \mathbb{Z}_{>1}$. Further, let us write $x = x_0 B^{n/2} + x_1$, and $y = y_0 B^{n/2} + y_1$, where $0 \leq x_0, x_1, y_0, y_1 < B^n$. The goal of this exercise is to reduce the multiplication of two integers of length n to a few multiplications of integers of lengths $n/2$.

(a) Show that

$$xy = x_1 y_1 B^n + (x_0 y_1 + x_1 y_0) B^{n/2} + x_0 y_0.$$

How many $n/2$-digits integers multiplications do you need to perform to compute xy?

(b) We denote by $T(n)$ the complexity of the multiplication of two n-digits integers using this approach. Show that T satisfies

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n).$$

(c) Using Theorem 1, can you solve this recursion formula for T? Is this approach asymptotically better than the naive approach?

(d) Karatsuba noticed that we can compute $(x_0 B^{n/2} + x_1)(y_0 B^{n/2} + y_1)$ using only 3 multiplications, instead of the four naive $x_0y_0, x_0y_1, x_1y_0, x_1y_1$.

More precisely, we let

$$\begin{aligned} z_0 &= x_0 y_0 \\ z_2 &= x_1 y_1 \\ z_1 &= (x_1 + x_0)(y_1 + y_0) - z_0 - z_2 \end{aligned}$$

Show that

$$xy = z_2 B^n + z_1 B^{n/2} + z_2.$$

(e) Write the recursion formula for the complexity $T(n)$ in terms of $T(n/2)$ with this new approach.

(f) Using Theorem 1, solve this recursion formula for T.

(g) Implement a recursive algorithm that uses Karatsuba's trick to compute the multiplication of any two integers in Magma. How does this algorithm compare to Magma's built-in multiplication for 2048-bit integers?