Algorithm Course Lab Report

Andre da Silva

Title: Karatsuba Algorithm for Multiplication

Date: January 26, 2025
Name: Andre da Silva
Student ID: adasi5

1 Introduction to the Lab's or Assignment's Content

Objective

For this lab report, I will be looking at the Karatsuba algorithm for multiplication, which is an improvement on the algorithm we all learned in elementary school.

Problem Statement

We will tackle the Karatsuba algorithm in this lab, which aims to reduce the complexity of multiplying 2 numbers from $O(n^2)$. Karatsuba managed to do this in $O(n^{log_3 2})$ instead. $log_3 2$ is about 1.6, so 1.6 is significantly better than 2, especially for very large inputs. These savings of about $n^{0.4}$ relative to the old algorithm will become enormously helpful as we let our inputs get as big as we like.

2 Lab Results

- **All Goals Achieved:** Yes. The recursion works as expected, so the results of the multiplication are correct, even when passing back and forth from strings to numbers. I tested Karatsuba's by hand– with small enough numbers– and saw how the algorithm worked, but not yet the efficiency gains because numbers that can be done by hand are too small to show those gains.
- **All Tests Passed:** Yes.

Note: Transparency is paramount. Specify any unmet goals or failed tests clearly and honestly.

3 Algorithm Framework

Algorithm Description and Key Steps

The Karatsuba algorithm divides the two numbers being multiplied in half (e.g., 34 = 3, 4; 2037 = 20, 37; 40367259 = 4036, 7259; etc.). The two parts of the top number are called ab, and the two parts of the bottom number are called cd. That is, the more significant half of the first number is A; and of the second number, C. And the less significant half of the first number is B; and of the second number, D

The correct answer for the product is obtained by recursive application of 10^{len}ac+$10^{len/2}$(ad+bc)+bd. If one of the two numbers in a multiplication is a single digit, do it the regular way; otherwise, use Karatsuba. "Otherwise, use Karatsuba" therefore makes the algorithm recursive.

For example, with 32875648*40367259 = an 8-digit by 8-digit multiplication, the grade-school way would require 64 multiplications.

Karatsuba, instead, requires splitting 32875648 into 3287 and 5648, and 40367259 into 4036 and 7259. The first multiplication Karatsuba will ask for is 3287 by 4036, eventually shifted by $10^{8/2}$.But instead of doing this out the regular way, Karatsuba's recursive nature means that the first step in this multiplication is to split 3287 into 32 and 87, and 4036 into 40 and 36. Then we need to do 32 by 40– but this still doesn't contain any single-digit operands. So we once again split the numbers, and get 3 and 2, and 4 and 0; only now do we actually do regular multiplication. This splitting is the key divide-and-conquer mechanism of the Karatsuba algorithm.

ChatGPT has helped me calculate that, using Karatsuba applied recursively, instead of the 64 place-value-by-place-value multiplications in the grade school way, we could get this done in 27 multiplications. This saves nearly 60% of the time, and these savings will only increase as the numbers get bigger.

The gap between Karatsuba's method and the traditional way is on the order of $O(n^{0.415})$– so with ChatGPT's help to write and execute (in ChatGPT) some Python code to generate a graph, I can see that, for example, a 1,000 by 1,000-digit multiplication requires about 1,000,000 multiplications the traditional way, but only about 57,000 if Karatsuba's method is used. Getting to do $\geq$ 93% less work for a problem of this size depending on which method is used is an incredible motivator to switch to Karatsuba's method for large numbers.

4 Complexity Analysis

Time Complexity (Worst Case)

$O(n^{log_2 3})$– this is not immediately obvious, but is given in the YouTube video by Dr. Bazett, and the proof requires the Master Theorem, etc., which we have not yet covered.

Space Complexity (Worst Case)

$O(n * log(n))$ from ChatGPT

5 Lessons Learned, Feedback, and Conclusion

Challenges Faced

Understanding what exactly the algorithm was– remembering the steps, seeing the recursion, etc.–and how it was advantageous over the grade school method was quite hard for the first day or two since I had never seen this way of multiplying before. But after watching Dr. Bazett's YouTube video and working through several examples– including making one up and going step-by-step in Word with a bulleted list to see how going into/coming out of the recursion works– it became much easier to understand.

Insights Gained

It is possible– and becomes increasingly more advantageous as the numbers grow larger/longer– to improve the algorithm for multiplying numbers which we learned in grade school, in which every value of every number in every place is multiplied by every other number. This algorithm works but is iterative and takes the most naive way through the problem. Applying a recursive divide-and-conquer approach based on splitting the number in half (a front half and a back half, the front half has more significant digits) and noticing some redundancies, it is possible to cut the computation time complexity for multiplication by quite a lot. This does require some overhead since the algorithm is divide-and-conquer, so it is not always more efficient than the traditional way. But after a certain point, the savings become noticeable, and Karatsuba will always be better. However, before that point, the overhead is such that it is better just to multiply the numbers in the normal way.

Addition is fundamentally faster than multiplication. Karatsuba achieves its savings by turning one of the four multiplications required to multiply a number ab by another cd (neither is necessarily just two digits, just split up into left and right halves) into an addition. Doing 3 multiplications (10 to some power times ac, the product bd, and 10 to another power times the sum (ad + bc)) saves time. Of course, the sum (ad+bc) requires two products.

But the genius of the Karatsuba algorithm is that ad, bc, and the other products on the way to the final answer may themselves be computed by Karatsuba's recursive method of splitting the number into halves and applying this method. Every recursion needs a base case, so we can say that there are either one or two trivial cases: when one or both numbers have only one digit, or when one of the numbers is a power of the base in which the computation is done. If the number is just a single digit, then we can default to the grade-school way. And if the number is a power of the base, then that is just a matter of shifting the number that specific amount to the left and filling the remaining space with zeroes: "x" times 1000, for instance, just appends "000" to whatever number x happens to be.

6 References and Use of Tools

Other References

The Fastest Multiplication Algorithm. YouTube, uploaded by Dr. Trefor Bazett, 8 May 2023, https://www.youtube.com/watch?v=frT1UPiJUOO.

"Karatsuba Multiplication"*. ChatGPT, *Day Month Year*, https://chatgpt.com/share/6796a14e-b560-8012-a420-a1a115b9f986.

"Random Large Numbers". ChatGPT, January 2025, https://chatgpt.com/c/6793ef88-2974-8012-947d-636a5ead1355.

"Multiplication Calculation Steps". ChatGPT, January 2025, https://chatgpt.com/c/6793ecc3-b518-8012-9260-2e9116a03d83.

"Karatsuba Multuplication Steps". ChatGPT, January 2025, https://chatgpt.com/c/6793bd0a-f580-8012-a520-abe1292447e7.

Note: Ensure all content is concise and to the point. Aim for two pages, but extend if necessary to ensure clarity and thoroughness.