4190.407 Algorithms-Homework 1: Part 2

[Programming Task: Efficient Integer Multiplication]

[due: 2025/09/22 10:59 PM]

Objective

Implement and analyze different algorithms for multiplying large integers. Each homework may be done in a group of at most two students. Groups may change for different tasks or homework assignments.

Task Description [40 points]

1. **Basic Multiplication**: Implement the traditional grade-school multiplication algorithm for multiplying two large integers. The function basic_multiply(x, y) will return the result of $x \cdot y$ as a string.
2. **Karatsuba's Algorithm**: Implement Karatsuba's divide-and-conquer algorithm for integer multiplication, which has a faster time complexity than the traditional approach. The function karatsuba_multiply(x, y) will return the result of $x \cdot y$ as a string.
3. **Comparison**: Write a program that compares the runtime of both algorithms for large integers (with at least 100 digits).

Requirements for Submission

- Each of the input integers x and y is assumed to be stored as a string.
- **Code Implementation**:
  - Submit the source code for both the traditional multiplication and Karatsuba's algorithm.
  - Code should be well-commented, clearly indicating the steps of each algorithm.
  - The comparison of the two algorithms should be automated in your code, with runtime measurements.
- **Performance Analysis**:

  - Run the algorithms on test cases of varying input sizes (e.g., 50, 100, 200 digits).
  - Include timing data in your output to show the performance of both algorithms.

- **Report**:
  - A brief report (1-2 pages) discussing:
    * Time complexity of each algorithm.
    * Test results, including charts/tables/figures that show runtime comparisons.
    * Which algorithm performed better and under what conditions.
    * Insights on why Karatsuba's algorithm is more efficient for larger inputs.

- **Bonus (Optional,10 points)**:
  - Research and implement the Toom-Cook algorithm and compare its performance with the other two methods.

Input and Output Example

Input file (`input.txt`):

```
123456789123456789
987654321987654321
```

Output file (`output.txt`):

```
121932631356500531347203169112635269
```

Submission Format

- All source code files: Students' code should be in a file (e.g., studentID_Task1.py in Python) with at least two functions: basic_multiply(x, y) and karatsuba_multiply(x, y).
- A README file (include students' names and IDs of the group) with instructions on how to compile and run your code.
- For each group: only one member should submit all the materials. Other members should submit a single text file indicating who is responsible for the submission.
- The report in PDF format.

Evaluation Criteria

- Correctness of implementation.
- Quality of the performance analysis.
- Clarity and thoroughness of the report.