Project

Data Structures and Algorithms

Adaptive Matrix Multiplication

In this project you will implement and analyze, empirically and theoretically, a method for multiplying square matrices. The method will adapt to use algorithmic techniques depending on the system architecture on which it is run.

***First ensure you are in a group of 4 people** and make a plan for who will do what and when your group will meet to follow up on the progress and discuss any problems or new ideas.*

Part 1

First read Chapter 4.2 on Strassen's method for Matrix Multiplication and D'Alberto and Nicolaus paper on adapting the algorithm to use depending on the computer architecture to use Strassen's algorithm and other methods for multiplying matrices [1].

Part 2

2a) Write **pseudocode** for Strassen's algorithm and the standard square matrix multiply algorithm that runs in $\Theta(n^3)$.
2b) Extend the pseudocode of Strassen's algorithm so that it will work with any value of n (not just n that is a power of 2).
2c) Give a complete a proof of the Asymptotic bounds for the runtime of Strassen's algorithm and square matrix multiply.
2d) Implement the standard Square Matrix Multiply, $\Theta(n^3)$, algorithm for matrix multiplication and Strassen's matrix multiplication algorithm.

Part 3

Conduct an experiment to estimate the constants in the runtime equations of these two algorithms. In this experiment you should use randomly generated square matrices of increasing size with values in the range [-1,1] . Give an estimate from your empirical results for the smallest value of n before Strassen's algorithm becomes more efficient than the Standard algorithm.

Part 4

Use the code from Part 2 and the result of Part 3 for n to implement a class that provides a simple adaptive method for matrix multiplication. The default value for the threshold for crossing over should be in a config file with default value set based on Part 3. The class should have at least the following 4 public methods:

- **void calibrate_crossover_point():** Estimates a value for the crossover point for matrix size to decide whether to use Strassen's approach or the standard approach by running a series of tests of multiplying randomly generated matrices of different sizes. It should save this value to a configuration file.

- **Matrix square_matrix_mutliply(Matrix a, Matrix b):** Use the standard approach to mutliply 2 matrices a and b, return their product.
- **Matrix strassen_mutliply(Matrix a, Matrix b):** Use Strassen's algorithm to multiply 2 matrices a and b, return their product.
- **Matrix adaptive_multiply(Matrix a, Matrix b):** For matrices that are less than the (previously stored or default) crossover point use standard matrix multiplication, for matrices that are greater than the crossover point use Strassen's method.

Provide an evaluation result extending the results from Part 3 to describe the performance of your implementation of the adaptive_mutliply method.

Additional Requirements for the matrix implementation:
The matrix implementation should be suitable for dense matrices. You are required to define a class Matrix that will be used in the implementation. This class Matrix will represent a matrix in "row-major" order (i.e. for an $n \times n$-matrix the first row will be stored in an array at index 0 to index n-1 the next row at n to 2n-1 and so on). The class should provide a constructor and methods to get and set the element at any row column index.

Deliverables
To complete the assignment you need to hand in the following deliverables.

1. **[40%]** The code for your implementation of the basic matrix multiplication algorithm and Strassen's algorithm. See also the requirements specifying the way you should represent matrices. The code should be well documented with each algorithmic step clearly commented (you can use the algorithm specifications from Chapter 4.1 and 4.2 in the textbook). This includes the code of part 4 and the additional requirements for matrix implementation.
2. **[30%]** A report including the following:
    a. A section titled ***background*** that is a 1-page summary of the contributions of the paper in [1]. Discuss why the authors of [1] think the topic of their paper is useful, a summary of the implementation of the adaptive method (how does it work?), and their experiment design and results. Also include some examples or particular situations where *you think* their results could be useful for the scientific community or industry.
    b. A section titled ***theoretical analysis*** that uses an abstract model to estimate the crossover point analytically. You can provide this based on a review of academic papers and the textbook but make sure you explain it clearly and discuss the structure, constants, assumptions, and limitations of the theoretical model.
    c. A section titled ***methodology*** in which you explain Strassens Algorithm, standard matrix multiplication and give **pseudocode** and further explanation. Include your runtime analysis from Part 2 here.
    d. A section titled ***experiment design*** *that* describes your implementation of the two algorithms and the way you generate test problems (ie matrices to multiply).
    e. A section titled ***empirical analysis*** that provides your results of parts 3 and 4 evaluating the adaptive method for matrix multiplication, Strassen's algorithm, and the basic method. You should use [1] as an example of the

type of results you should put in this section (tables, graphs, type of discussion) because when marking will expect to see graphs and results that are of comparable quality to this and measure similar quantities.

   f. The report should also have an ***introduction*** and ***conclusion***.

3. **[20%]** A presentation and power point slides.
4. **[10%]** Project management. This should include a clear work break down structure in which you have divided the work into small tasks that will each take at most 3 days of work. It should also include a summary (minutes) of each of your meetings and the assignment of the work task. The description of the Parts 1-4 should give you a start on the work breakdown structure. It is possible for more than one person to work on some tasks (e.g. pair programming could be used for some complex tasks if they cannot be broken down into further smaller tasks).

References:

[1] Paolo D'Alberto and Alexandru Nicolau. Adaptive Strassen's matrix multiplication. In Proceedings of the 21st Annual International Conference on Supercomputing, pages 284– 292, June 2007