

---

Programming Assignment # 1Due Sunday, Sept. 25th, 11:59pm

---

Strassen's divide and conquer algorithm is asymptotically faster than the conventional  $O(n^3)$  algorithm for matrix multiplication. This means that for sufficiently large values of  $n$ , Strassen's algorithm will run faster than the conventional algorithm on instances of size  $n$ . However, for small values of  $n$ , the conventional algorithm is faster.

Let us define the *cross-over point* between the two algorithms to be the smallest value of  $n$  beyond which the asymptotically faster algorithm actually runs faster. The goal of this assignment is to study the cross-over point between Strassen and the standard algorithm, both analytically and experimentally, as well as to understand some factors that can dramatically affect the actual speed achieved by an implementation of an algorithm.

1. Assume that the cost of a single arithmetic operation (adding or multiplying two numbers) is 1. Use this assumption to derive simple and crude expressions for the running time of Strassen's algorithm and the standard matrix multiplication algorithms (including the constants). Use your results to derive a crude estimate of the cross-over point for the two algorithms.
2. Implement Strassen's algorithm and the standard matrix multiplication algorithm to find the cross-over point experimentally. The actual cross-over point you obtain will depend upon how carefully you implement Strassen's algorithm. A couple of important considerations in that implementation include:
  - (a) keeping track of the recursion explicitly using your own stack,
  - (b) stopping the recursion at some  $n_0$ , and using the conventional algorithm below that point.

Try to make the cross-over point as small as you can.

You may work **in pairs**. You will submit your assignment, **both a report and code via WebCT**. The grading policy would be 60% for the report and 40% for the code.

---

### Submission guidelines:

1. Input and output file should be named **input.dat** and **output.dat** respectively. See class website for input/output format. First line gives the value of  $n$ , followed by two matrices  $A$  and  $B$  of size  $n \times n$ .
2. Use of libraries for matrix operations (like multiplication or addition etc..) is not allowed
3. Programming language should be **C/C++/Java**.
4. Make sure everything is in one report (including README).
5. Mention the platform (linux/Win32/ etc...) on which the program should be tested in the report.
6. Include a Makefile. Name of the executable should be **mat\_strassen** and **mat\_standard** for strassen's and standard algorithm resp. If using win32, then give instructions for compiling and running the program.