

High Performance Computing Lab (CS 530):

Lab 1: Implement the power of a square matrix. Given a large square matrix A (with a dimensionality M), and a number N , implement the power operation A^N , and assess the running complexity of the algorithm. Implement a version in C/C++ for a) 2D arrays (classical solution) and b) double linked lists. The classical version should also be implemented in one script language of your choice such as Python, R, Matlab, Octave, Ruby, etc.

Overview

Matrix multiplication is an easy task to accomplish, however, the complexity of the multiplication is polynomial, and for large matrices such an operation can take a very long time.

Instructions

- Given a square matrix A , and a positive number N , the N^{th} power of the matrix A is defined as follows: $A^N = A^{N-1} \times A$.
- The dimensionality M of the matrix A should be provided from command line.
- The N parameter should also be provided from the keyboard.
- The program should work for large matrices, therefore dynamic memory allocation should be considered. For extremely large matrices list structures should be considered.
- Due to the size of the matrix, the elements should be populated using random numbers or with an option for the identity matrix.
- Calculate the time complexity of the operations.
- Plot running complexity graphs for different M sizes and different N values using all three solutions. All kind of plots are accepted (Matlab, R, Excel, Octave, Python, etc.). Ex. $M = \{1, \dots, 1000\}$
 $N = \{0, \dots, 10000\}$

Notes

- Error handling is expected.
- The program should work for large matrices.
- The program(s) should be written in C/C++ and one script language of your choice.
- No special matrix related library should be considered.
- Already existing data structures can be considered to model the matrix.
- Start multiple jobs, and see how the complexity changes.
- The plots should be inserted in a short report describing the problem, the solutions and the explanation of the graphs. The format should be pdf and generated using LaTeX. The document style should be report. The name convention: FirstName_LastName_Lab1.pdf.
- Compare different time functions: `gettimeofday()`, `time()`, `clock()`, etc.

Rubric

Task	Points
Error handling	1
A^N for large M and N values in C/C++ and another script language	2 (classical in C/.C++) + 2 (classical in a script language) + 2 (dynamic lists)
Report	2
Comparison of different time functions	1