# CSC 191B: Lab #3: Growth Curves

---

**Learning Outcomes**

- Perform operations on 1D arrays
- Plot functions and set plotting options and formatting
- Use experiments to reason about asymptotic behavior of functions

---

**Background.** A central question in computer science is: "How long will it take for my program to finish?" In particular, we'd like to know the time required as a function of the size of the input to the program. For example, we'd expect a text analysis algorithm to process one newspaper article a lot faster that it takes to analyze all the New York Times articles written in a given year. We usually call this function of the input size the *time complexity* of an algorithm.

Searching through an unsorted array of numbers requires accessing every element in the array, so the time complexity of that algorithm applied to an array of size $n$ is approximately $f(n) = n$. If the array is sorted, then binary search can be done with time complexity approximately $f(n) = \log_2(n)$, which is much much faster than linear search when $n$ is large.

A logarithmic time complexity is a desirable trait of an algorithm – it means the algorithm won't take long to run even for big inputs because the function grows so slowly. A surprising fact is that $\log(n)$ grows more slowly than $n^\epsilon$ for *any* $\epsilon > 0$. That is, given any $\epsilon > 0$, such as 1/4 or 0.000001, there exists an $N$ such that $\log(n) < n^\epsilon$ for all $n > N$. This is true of logarithms for any base, but for this lab, $\log n$ refers to the natural logarithm, $\ln n$ or $\log_e n$, which corresponds to MATLAB's `log` function.

**Problem Statement.** In this lab, we'll use MATLAB's plotting functions to visualize growth curves and understand their behavior.

1. Plot the functions $f_1(n) = n$, $f_2(n) = n^2$, $f_3(n) = n^{1/4}$, and $f_4(n) = \log n$ on the same axes over the range $n \in [1, 7]$. Include a title, axes labels, and a legend with your plot, and find ways to make the plot presentable by adjusting `FontSize` or legend `Location`, for example. Useful MATLAB functions for this task include `linspace`, `.^`, `log`, `plot`, `hold on`, `legend`, `title`, `xlabel`, and `ylabel`.
2. Use MATLAB's plotting tools to find when $n^{1/4}$ overtakes $\log n$. Plot the two functions in the range that shows the crossover point using `logspace` and `semilogx`. (Hint: it's a big number.) Include a title, axes labels, and a legend with your plot, and find ways to make the plot presentable by adjusting `FontSize` or legend `Location`, for example. Use plot symbols (such as `'*'` and `'o'` to demonstrate that the logarithmically spaced $n$ values look linearly spaced on a logarithmic x-axis.

**Discussion.**

1. What techniques did you use to find the crossover point between $\log n$ and $n^{1/4}$? What is the crossover point between $\log n$ and $n^{1/5}$? (You need only provide the number here.)
2. What plotting options did you use to make the plots presentable? Why did you choose those? What else did you try?

**What to turn in.**

- One MATLAB file `polylog.m` that contains the commands to generate the two plots and commented lines that answer the discussion questions
- Two pdfs that contain the MATLAB-generated plots (don't attached .fig files, use File/Save As on the MATLAB figure window)

---

**Grading rubric**

- Code: 40 points each for MATLAB commands to generate the plot (they must be well commented) and the plots themselves
- Results: 10 points each for the discussion questions