Lecture 1&2 Asymtotic Notation

- **Algorithm Analysis**
  - Primary interest
    - Time Complexity
      - Big O Notation
        - Gives an upper bound on the growth rate of a function
  - Secondary interest
    - Space Complexity
      - A measure of the amount of working storage an algorithm needs
      - How much memory the algorithm needs
- **Primitive Operations**
  - Single addition
  - Multiplication
  - Comparison
- **Common Functions**
  - Constant
    - O (1)
  - Logarithmic
    - O (log n)
  - Linear
    - O (n)
  - Log-linear
    - O (n log n)
  - Quadratic
    - O (n^2)
  - Cubic
    - O (n^3)
  - Polynomial
    - O (n^k)
  - Exponential
    - O (a^n), a > 1
  - Factorial
    - O (n!)
- **Big Omega and Theta Notation**
  - Big O
    - f(n) is O(g(n)) if f(n) is asymptotically less than or equal to g(n)
  - Big Omega
    - f(n) is Ω(g(n)) if f(n) is asymptotically greater than or equal to g(n)
  - Big Theta
    - f(n) is Θ(g(n)) if f(n) is asymptotically equal to g(n)