Analysis of Algorithms

Tuesday, February 13, 2018 12:12 PM

Algorithm Complexity

- Time and Space resources (especially for large problems)
    - Small problems - differences are insignificant
- Emphasis is usually on time (vast improvements in memory space)
- Not about comparing the exact run time
    - Different languages/ data sets/ computers/ implementation
- Worst, Best and Average case analysis

Measures

- Execution Time: number of primitive operations/ statements executed
- Growth Rates: a function of problem size

Asymptotic Analysis

- Large input size, considers leading term sans coefficient
- (Asymptotic) upper bound = O(g(n)) [tight bound]
    - g(n) = big-O notation of f(n)
    - Show with c (coefficient) and n_0 (minimum n size)
    - Growth term, order of magnitude

Complexities

- O(1) - constant
- O(log n) - logarithmic (reducing the problem by a fraction)
    - Base of logarithm does not matter: constant coefficient
- O(n) - linear
- O(n log n) - polylogarithmic (usually sorting)
    - Tractable limit
- O(n^2) - quadratic (nested loops)
- O(2^n) - exponential
    - O(2^n) vs O(3^n) related by a constant
    - Tower of hanoi proof