Tutorial Session

Algorithm & Counting methods

Pseudo Code of Algorithms

- Algorithms are recipes to solve problems
    - Finite, precise
- For, while, if … then …, if … else if … then …
- Recursive algorithms
    - A routine that calls itself (with a reduced input)

Algorithmic Complexity

- Measures the # of basic operations
  - A function of input size
- Asymptotic notation (Big-O, Big-Ω , Big-Θ, small-o, small-ω)
  - Definitions
  - Finding the dominating terms
  - Write functions in forms of the asymptotic notations and compare their complexity

Big-O definition

DEF: Let f, g be functions with domain $\mathbf{R}_{\geq 0}$ or $\mathbf{N}$ and codomain $\mathbf{R}$. If there are constants C and k such

$$\forall\ x > k,\ |f(x)| \leq C \cdot |g(x)|$$

then we write:

$$f(x) = O(g(x))$$

- Big-Θ: f(x) = O(g(x)) & g(x) = O(f(x))

Rule of thumbs

- First, for input size n, determine the # of basic operations as f(n)
- Find the dominating term in f(n)
- The following functions are in growing order of complexity

$$\frac{1}{x}, \ln x, x, x^{e}, e^{x}, x^{x}$$

Counting methods

- Multiplication principle
  - Count in stages
- Addition principle
  - Divide the original set into disjoint sets
- Inclusion-exclusion principle
  - Generalization of the addition principle to overlapping sets
- Pigeon hole principle
  - Given N pigeon, k holes, at least one hole contains $\lceil N/k \rceil$ pigeons
  - Can also solve the inverse problem, how big N needs to be such that for k holes, at least one hole contains $\lceil N/k \rceil$ pigeons