Announcements

• Homework 4 due date changed
• Vipul will lead a review session next class
• Final exam December 20
• Do class evaluations
Outline

• Program Analysis
  – complexity measurements

• Answer the question:
  – how do our algorithms (programs) behave in general?

• Overall considerations
Computer Science Theory

- **Computability**
  - whether or not a problem is solvable
  - classify problems as solvable or not

- **Complexity**
  - whether or not a problem is difficult to solve
  - classify problems as easy or hard
The Goal

• Programming is a problem-solving activity

• Even if a problem can be solved:
  - measure the quality of the solution
  - time requirements
  - space requirements

• You see this tradeoff in HW4
HW4

- Brute force searching
- Saving the “right amount” of intermediate transformations to amortize the cost in future runs
- In-memory vs. on-disk dictionary
  - data structure/representation
  - fast access in one, large space in another
Measuring Time Complexity

• Amount of work done measured as a function of input size (N)
  – for a graph, may be size of V + E
  – for a string, the length

• “Number of steps” or basic operations
Cases

• **Worst case**

• **Best case**

• **Average case**

• **Example: linear search**
Expressing Complexity: Big-O

• The number of steps is usually a complex expression

• So we estimate
  - asymptotic analysis
  - running time on large inputs
  - consider only the highest order terms

• $f(n) = 2n^2 + 5n^3 + 7n + 89$
Big-O Definition

• Let \( f \) and \( g \) be 2 functions \( f, g : \mathbb{N} \rightarrow \mathbb{R}^+ \)

• Say that \( f(n) = O(g(n)) \) if positive integers \( c \) and \( n_0 \) exist so that for every integer \( n \geq n_0 \):
  \[
  f(n) \leq cg(n)
  \]

• Thus, \( g(n) \) is an upper bound for \( f(n) \)
Example

• \( f(n) = 5n^3 + 6 \)
  Let \( c = 6 \) and \( n_0 \) be 10

• if \( g(n) = cn^4 \), this is still valid. We can be arbitrarily large
The PATH Problem

• Given $G = (V, E)$
  - is there a path from $s$ to $t$?
  - brute force is $m^m$, where $m$ is sizeof($V$)

• Solution:
  - BFS: mark nodes reachable from $s$ at distance 1, distance 2, distance 3, etc.
  - solvable in polynomial time (P)
Hard Problems

• Cryptography
  - message decoding should be exponentially hard in the size of the input
  - a key is combined with a plaintext to produce ciphertext