WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

Cpt S 122 – Data Structures

Course Introduction

Nirmalya Roy

School of Electrical Engineering and Computer Science
Washington State University

Course Introduction

- Course Information
  - Who am I?
  - Teaching Assistants
  - Course Logistics
  - Evaluation and Grading
  - Assignments
  - Exams
  - Course Expectations
- Course Overview
- Questions

Who Am I?

- Dr. Nirmalya Roy
  - MS in CSE: UT-Arlington, 2004
  - PhD in CSE: UT-Arlington, 2008
  - Postdoc in ECE: UT-Austin, 2010
  - Research Scientist at Institute for Infocomm Research (I2R), Singapore, 2010-2011
- Email: nroy@eecs.wsu.edu
- Homepage: http://eecs.wsu.edu/~nroy/
- Office: EME 127
- Office Hours:
  - Wednesday and Friday 2:00 pm – 3:00 pm

Teaching Assistants

- Chris Cain
- Shervin Hajiamini
  - Office Hours: Tuesday 1:00 - 2:00pm
  - Location: Sloan 343
- Matt Hintzke
- Rachel King
- Evan Olds
  - Office Hours: Thursday 12:00 - 2:00pm
  - Location: Sloan 353
- Jey Salem

Course Logistics

- Time and Location
  - Monday and Wednesday and Friday 12:10pm - 1:00pm
  - TODD 216
- Course description
  - In this course, we use the C/C++ programming languages to explore the fundamental concepts, constructs, and techniques of modern computer programming, including data structures, software engineering, and classes and objects. The primary aim of this course is to refine your problem solving and programming skills so that you may apply efficient data structures to real engineering problems.
- Course website
  - www.eecs.wsu.edu/~nroy/courses/cpts122/
- Prerequisites
  - Cpt S 121 (Program Design and Development) or an equivalent course.

TextBook

- Required TextBook:
  - *P.J. Deitel & H.M. Deitel, C: How to Program (7th ed.), Prentice Hall, 2012. ISBN: 9780132990448*

- Reference Textbook:
  - *Accelerated C++: Practical Programming by Example, 2000 by Andrew Koenig and Barbara E. Moo*

Evaluation and Grading

- Evaluation
  - 5 -6 Quizzes: 5% of grade
  - 7-8 Programming Assignments: 35% of grade
  - 2 Midterm Exams (10% per midterm) and Final Exam (20%): 40% of grade
  - 13 Labs: 20% of grade
- Grading Scale
  - 90-100%: A
  - 80-89%: B
  - 70-79%: C
  - 60-69%: D
  - 0-59%: F

Assignments

- Quizzes (# 5 to 6)
  - 9-10 objective questions
  - delivered mostly at the end of the class
- Programming Assignments (# 7 to 8)
  - 8 programming assignments
  - due through angel dropbox generally two weeks later **electronically by 11:59pm** (mechanism to be described in advance of the first assignment)
  - **No late assignments will be accepted**

Labs

- Labs (# 13)
  - 13 lab assignments for 13 weeks
  - should be done and due in the lab section you are enrolled with
- Lab Location: Sloan 353W (check the WSU building description for more information about the location)
- Lab Times:
  - Section 01: TU 12:00 – 2:50 pm; TA: Evan Olds
  - Section 02: TU 9:10 – 12:00 pm; TA: Jey Salem
  - Section 03: TU 2:50 - 5:40 pm; TA: Chris Cain
  - Section 04: TU 5:40 – 8:30 pm; TA: Shervin Hajiamini
  - Section 05: W 9:10 - 12:00 pm; TA: Matt Hintzke
  - Section 06: TH 5:40 – 8:30 pm; TA: Rachel King
- Required Software: Microsoft Visual Studio 2008 (for the programming assignments); Microsoft Visual Studio is designed for Windows machines only.
- Microsoft Visual C++ 2008 Express Edition
- VMWare Fusion to run Windows in Mac

Exams

- **Tentative exam dates:**
  - September 28, 2012
  - November 5, 2012
  - Final 12th December (Wednesday), 2012 in the Classroom 1:00 to 4:00 pm
- All exams are cumulative

Course Expectations

- Attendance
  - You should attend class. Lecture notes will be made available, but they should not be considered a substitution for attending class
- Collaboration
  - You can discuss both programming assignments and labs with other students at a conceptual level
  - Do **not write or program while talking to a fellow student**
  - Do **not use any other resources without citation**

Course Overview

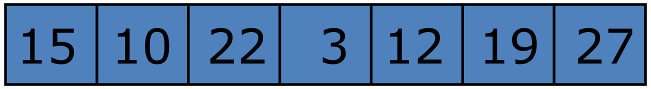
- C Review
  - Functions, Recursion, Pointers, Characters and Strings
- Data structures
  - Linked Lists, Stacks, Queues, Tress
- Overview of C++
  - Classes and Objects, Operator Overloading, Inheritance, Polymorphism, Template, Exception Handling
- Abstract Data Types
  - Linked Lists, Stacks and Queues
  - Insert, delete, search, sort

Course Overview

- Advanced data structures
  - Sorting, Hash tables
- Algorithm development and analysis
  - Running time

Data Structures

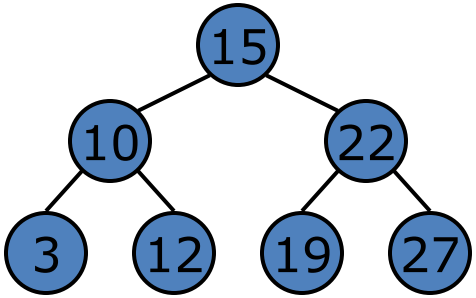
- “Why not just use a big array?”
- Example problem
  - Search for a number k in a set of N numbers
- Solution # 1: Linear Search
  - Store numbers in an array of size N
  - Iterate through array until find k
  - Number of checks
    - Best case: 1 (k=15)
    - Worst case: N (k=27)
    - Average case: N/2



15	10	22	3	12	19	27

Blessing of Data Structures

- Solution # 2: Binary Search Tree (BST)
  - Store numbers in a binary search tree
    - Requires: Elements to be sorted
  - Properties:
    - The left subtree of a node contains only nodes with keys less than the node's key
    - The right subtree of a node contains only nodes with keys greater than the node's key
    - Both the left and right subtrees must also be binary search trees
  - Search tree until find k
  - Number of checks
    - Best case: 1 (k=15)
    - Worst case: $\log_2 N$ (k=27)
    - Average case: $(\log_2 N) / 2$

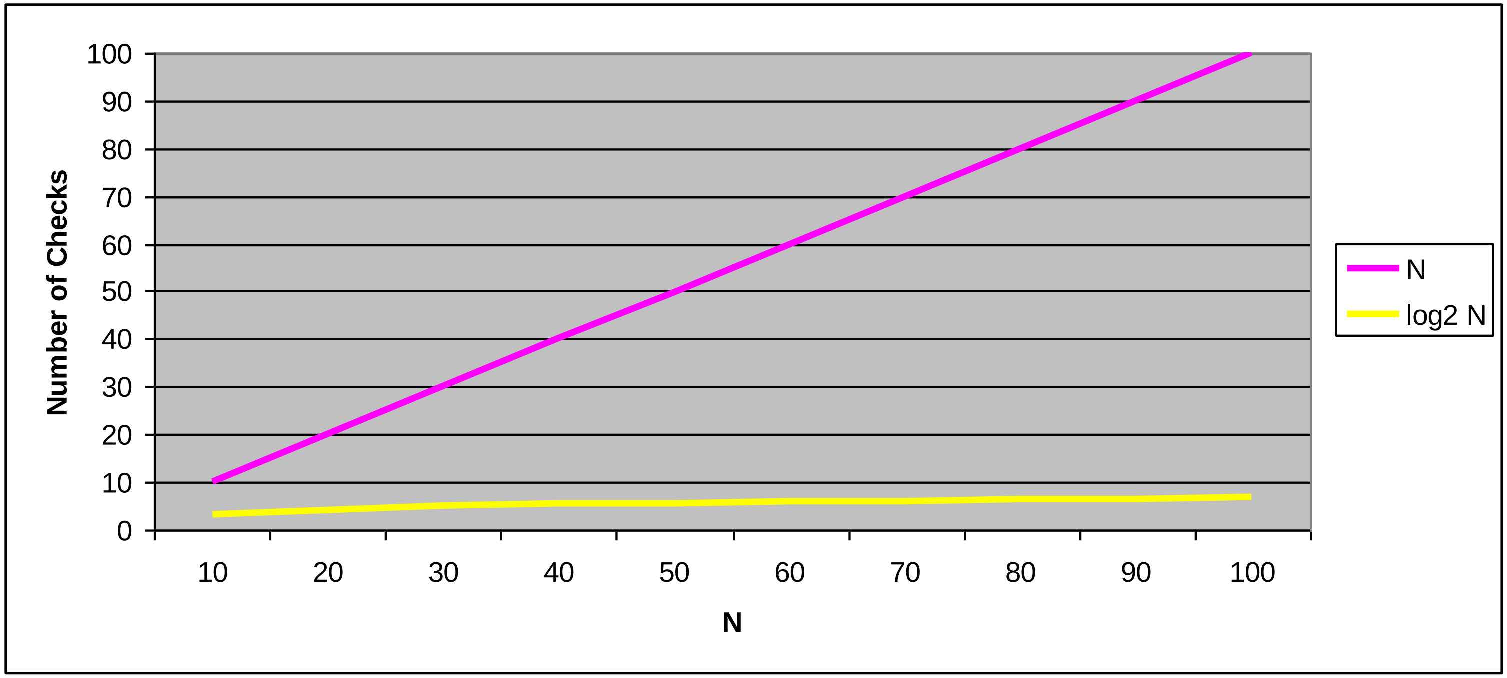


Example

- Does it matter?
- Assume
  - N = 1,000,000,000 = 10^9
    - 1 billion (Walmart transactions in 100 days)
  - 1 Ghz processor = 10^9 cycles per second
- Solution #1: Linear Search (10 cycles per check)
  - Worst case: 1 billion checks = 10 seconds
- Solution #2: Binary Search Tree (100 cycles per check)
  - Worst case: 30 checks = 0.000003 seconds

Analysis

- Does it matter?
  - N vs. ($\log_2$ N)



Insights

- Moral
  - Appropriate data structures ease design and improve performance
- Challenge
  - Design appropriate data structure and associated algorithms for a problem
  - Analyze to show improved performance

Next Class—Function Review

- Program Modules
- Function Definitions
  - Prototypes
  - Function Calls
  - Data Structures in Function Calls
- Passing arguments by value and by reference

Questions

?