CPSC 259: Data Structures and Algorithms for Electrical Engineers

Introduction

Hassan Khosravi

Learning Goal

- Get an overview of the course

Course webpage

- The course home page

  http://www.ugrad.cs.ubc.ca/~cs259/2015W1

- Read it carefully
  - Course info
  - Notes(Pre-lecture notes, Post-lecture notes)
  - Labs (A lot of info)
    - Labs start on Monday, September 14.
    - Exercises
  - Info on the teaching team and their Office hours

Connect

- labs, and your grades are hosted on Connect
- To access Connect you need a Campus-Wide Login (CWL), Netinfo account or Student Number and PIN. To register for a CWL account, visit www.cwl.ubc.ca.
- Course textbook
  - Data Structures Using C (second edition)
    - First edition should also be fine

Piazza

- The course bulletin
  - http://piazza.com/ubc.ca/winterterm12014/cpsc259/
  - Enrollment code (cpsc259)
  - The Piazza bulletin board is required reading. It will be used for important material that may not be mentioned in class.
  - Problems with the CPSC 259 course contents (e.g., lecture, lab, textbook, assignments) can be posted on Piazza,
  - Students are encouraged to ask questions, and to respond to other students' questions
  - Only send email if you can't post the material on Piazza

- Selected individuals that actively respond to postings on Piazza, may receive a bonus mark of up to 1%.

Exams

- Midterm 1
  - Date: Wednesday, October 21
  - Time: 08:00 - 08:50
  - Location: Hebb - Room 100
- Midterm 2
  - Date: Friday, November 13
  - Time: 08:00 - 08:50
  - Location: Hebb - Room 100
- If you have a conflict let me know ASAP and no later than Monday, September 14

Labs

- There are 5 labs(two weeks each)
  - **in-lab component:**
    - During Week 1's lab time an in-lab component
    - Completed in pairs, has deliverables that are due either during your Week 1 lab time -or, for some deliverables, no later than during Week 2's lab time
  - **in-lab programming test:**
    - During Week 2's lab time, The first hour will be a programming test, which is to be done individually.
    - You will use a computer and the Visual Studio 2012 C/C++ compiler to implement some code. The style is similar to that in APSC 160.

Labs

- There are 5 labs(two weeks each)
  - **take-home programming assignment:**
    - During the two-week stretch (minus a day) starting with Week 1's lab time, a take-home programming assignment to be completed with your in-lab partner.
    - For example, if your lab time is on Monday, then your "two weeks" end just before midnight on Sunday (about 13.5 days from the start of Week 1).
    - The last 50 minutes of lab time in Week 2 will allow you and your partner to work on your take-home assignments. TAs will be in the lab to help.

Clickers

- Starting this Friday you will be marked for your participation using i>Clickers during lectures.
- Be sure that you have registered your i>Clicker using UBC Connect before the second lecture.

Planned (tentative) Assessments

- 3% for the clicker questions (participation only)
- 4% for PeerWise
- 5% for the in-lab component, teamwork, and deliverables.
- 10% for the 5 in-lab programming tests (2% each).
- 15% for the take-home programming assignments.
- 28% for the midterm exams.
- 35% for the final exam

○ 1% bonus for active participation on Piazza

○ 1% bonus for PeerWise

○ To pass the course, you must obtain at least a 50% overall course mark (as per the above formula) and you must pass the final exam.

Prerequisites

- APSC 160
- If you don’t have the prerequisites you will be dropped from the course. Check with a CS Advisor if you have a problem
- The instructor cannot change your lab section or add you to the course.
  - Check with CS front office

Academic Concession

Students registered with Access & Diversity must

- inform the instructor within seven (7) days of adding the course or the start of term (whichever is later)(email me please)
- provide a copy of the letter from A&D granting accommodations (by email)
- submit an accommodation form at least two weeks prior to each exam for which accommodation is requested

Course Policy on Plagiarism

- Although you may discuss problem solving approaches to homework with other students, and get help from the TAs, you and your partner (as a team) must do your own work.
  - Plagiarism occurs when you submit someone else's work as if it were your own. Examples of plagiarism include:
    - copying some or all of another team's code; hiring a tutor to write some of the code for you; sending someone (other than your partner for that lab) your code via e-mail, text message, Skype, Facebook, Microsoft Messenger, etc.
  - The penalty for plagiarism can be a suspension from the University for 4-12 months and a grade of zero in the course. So, please don't turn in someone else's work.

Visual Studio

- You can get your own copy:
    - Engineering students can download a free copy of Microsoft's Visual Studio Professional using the ECE or CPSC department's DreamSpark academic licensing arrangement (formerly the MSDNAA agreement).
        - Details are written up in Lab 1

Okay if you prefer to use another IDE, but you will use Visual Studio in labs, and you should always be sure that your submitted code compiles and runs under Visual Studio.

Running Visual Studio on your laptop

- Running Visual studio on a Mac is a little tricky. You need to run Windows as a virtual machine.
  - Get virtual box https://www.virtualbox.org/
  - Get an iso image of Microsoft Windows 7 from https://www.cs.ubc.ca/support/available-software-dreamspark
  - Create a virtual Microsoft Windows 7 through Virtual Box.
  - Download and install Visual Studio on the virtual machine.
- Attend TA's office hours this week to get assistant for running visual studio on your laptop.

Learning goals

- Learning goals for each chapter will be given
  - Please pay attention to them.
  - They' re part of the lecture slides.
  - They make great exam checklists.

A word on using your computer during class…

Multitasking on your computer during CPSC 259 lectures:

A. Helps me to learn CPSC 259 materials because I’m not bored and has no effect on other people

B. Helps me to learn better, but distracts the people around me

C. Makes me learn worse, but has no effect on other people

D. Makes me learn worse, and distracts the people around me

Students’ use of laptops in class lowers grades: Canadian study

- “all the participants used laptops to take notes during a lecture on meteorology. But half were also asked to complete a series of unrelated tasks on their computers when they felt they could spare some time. Those tasks — which included online searches for information — were meant to mimic what distracted students might do during class.”
- The students who were asked to multitasked averaged 11% lower on the exam
- Their neighbors also did significantly worse than the average.

So, you already know how to program? Great, what's left for us to do?

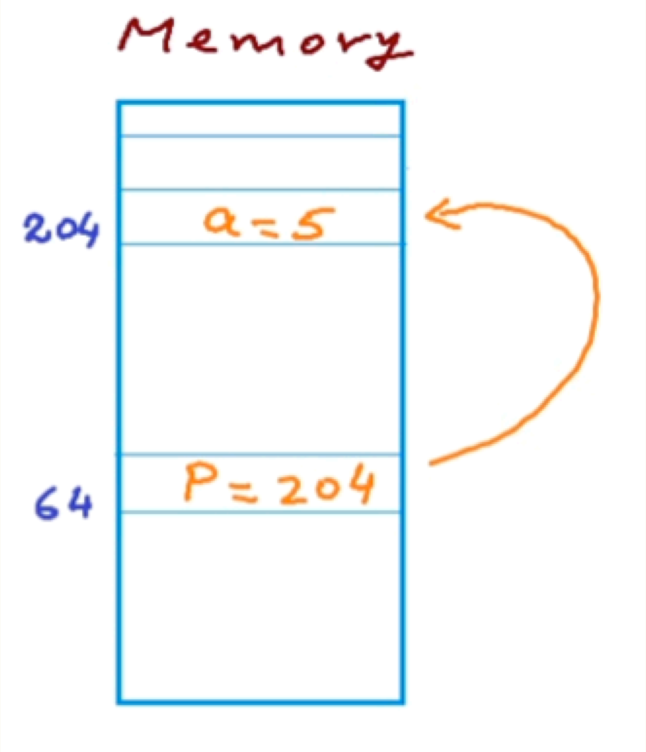
- In this course we cover the following material

1- APSC 160 review

- Do you actually remember how to program?
  - Briefly review some key programming concepts from APSC 160 (functions, modularity, arrays)

2- Pointers

- An object variable contains an object, but a pointer specifies where an object is located.
  - ○ Pointers can refer to objects that are allocated on demand.
  - ○ Pointers can be used for shared access to objects.
  - ○ Pointers are used in many complex data structures.



3- Structs (Records)

- We have used primitive data types
  - int, float, string…
- What if our program required more complex data types?
  - Social network applications require use of data types such as users
    - Age
    - Location
    - List of friends

  - We'll see how we can create our own data types such as users.

4- Complexity

- Let’s assume that I have three functions that all do the same thing.
  - How can I compare these algorithms?
  - Are there tradeoffs between the algorithms?
  - Based on a specific set of inputs can I conclude that one algorithm is better than the other algorithms.

5- Linked lists

- A linked list is a data structure for collecting a sequence of objects, such that addition and removal of elements in the middle of the sequence is efficient.
  - Removal of a user from a network.
  - Adding a new item into a sorted list

- This is not the case with arrays.

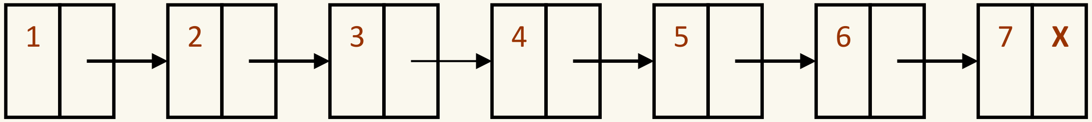
- We'll use pointers to build linked lists.

5- Linked List

- Linked list data structure in which elements can be added to or deleted from anywhere at will.
- In a linked list, each element, called a node, is allocated space as it is added to the list. Every node in the list points to the next node in the list. Therefore, in a linked list every node contains two types of information:

The data stored in the node and

A pointer or link to the next node in the list



6- Recursion

- Recursion is a powerful technique to break up complex computational problems into simpler ones. The term “recursion” refers to the fact that the same computation recurs, or occurs repeatedly, as the problem is solved.
- 5! = 5*4*3*2*1
- 5!= 5*4!

7- Binary Search Tree

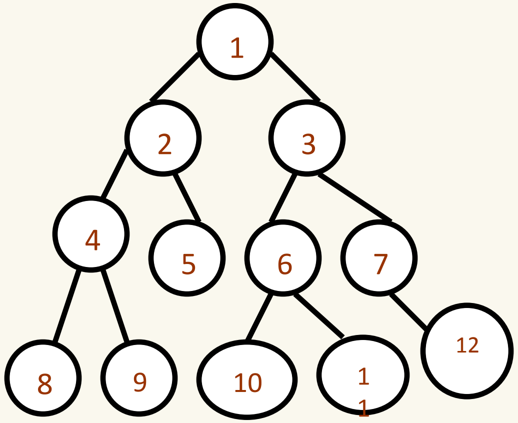
- In APSC 160 we learned how to use an array to store some items and then search for them.

12	**34**	**89**	**18**	**21**	**-1**	

- Can we use a different data structure to find items more efficiently.

7- Binary Search Tree

- A binary tree is a data structure which is defined as a collection of elements called nodes.
- Every node contains a "left" pointer, a "right" pointer, and a data element.
- Advantages: Provides quick search, insert and delete operations
- Disadvantage: Complicated deletion algorithm



8-Binary Heaps and Priority Queues

- Let’s say we have the following tasks.

  2 - Water plants

  5 - Order cleaning supplies

  1 - Clean coffee maker

  3 - Empty trash

  9 - Fix overflowing sink

  2 - Shampoo carpets

  4 - Replace light bulb

  1 - Remove pencil sharpener shavings

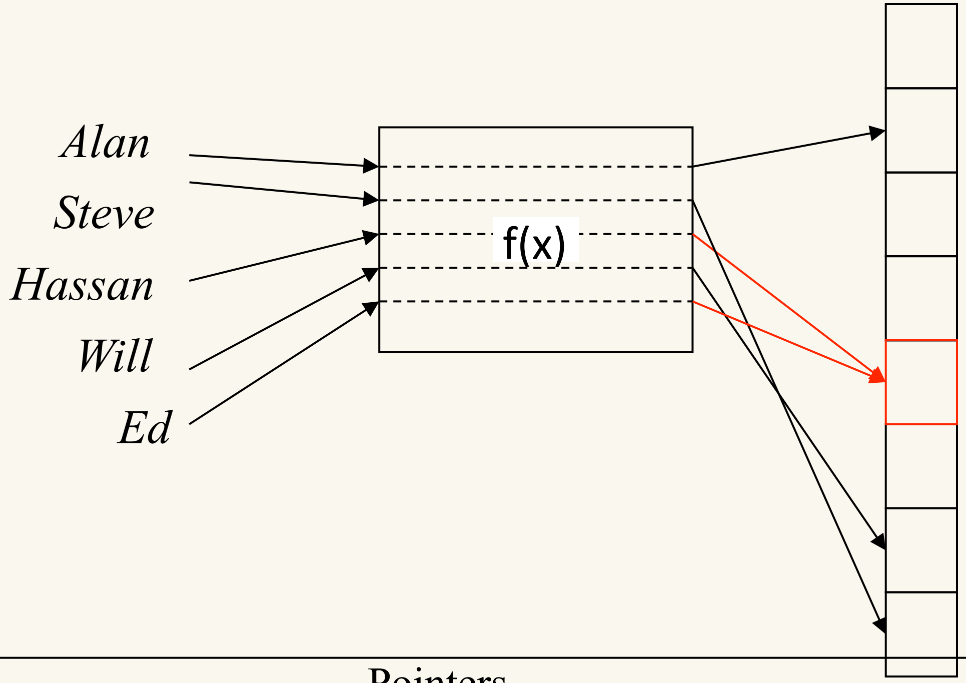
- What kind of data structure can we use to help us with this problem?

9-Sorting

- How can I sort 1 million values?
  - Is there one algorithm that is suitable for all different cases?
    - What are some of the well known algorithms?
    - What is the complexity of each of these algorithms?

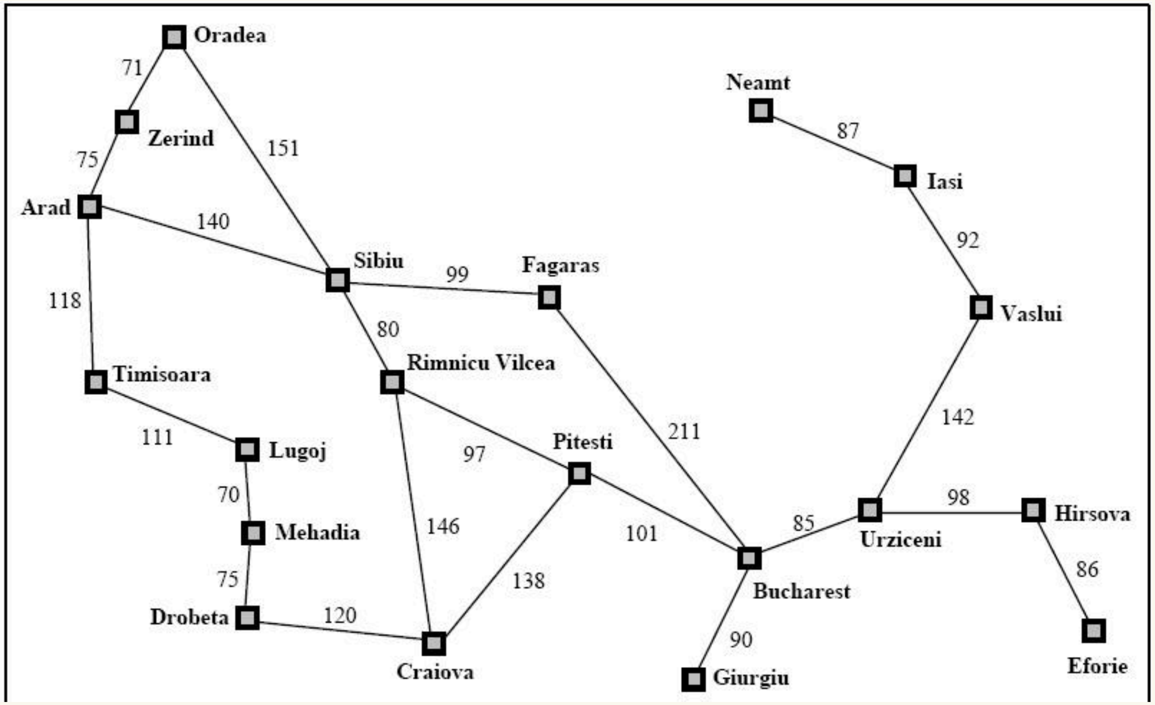
10 - Hashing

- In APSC 160 we learned how to use an array to store some items and then search for them.
- Hash function: maps keys to integers
  - result: can quickly find the right spot for a given entry



11- Graph Theory (potentially)

- Many real world problems can be shown by graphs



- What data structure is suitable for representing a graph?
- What is the shortest path between two cities?