Connecting with Computer Science

Defining Information Technology & Data Organization

Data Types and Computer Basics
Administrative notes

- Reading Questions 1 and 2 are up
  - RQ1 is due Sunday at noon
  - RQ2 is due Wednesday at noon
  - In general I’ll try to have them up a week ahead of time
- Connect has been populated – you should be able to see it now
- Labs will be posted each Friday the week before you will do it in lab.
Feedback on RQ0

• Some students gave the answers to the exercises given at the end of the reading.
• Omit needless words.
• Submit only one question.
• Don’t ask a question that you can easily search for yourself (e.g., “are the websites in the textbook real”)
Learning goals

• Understand what an algorithm is and why they matter
• Give an example of cases when one algorithm might perform better or worse than another algorithm
• Recognize examples of **data types** and illustrate how properties associated with familiar data types can influence the behaviour of computer applications which act on these data types
• Recognize examples of **data structures** and classify data structures as networked, hierarchical, and/or tabular when applicable
• Convert between small instances of graphical and text based representations of hierarchical structure
Hardware vs. Software

Last time we discussed what a computer is. One thing we didn’t differentiate is hardware vs. software.

**Hardware** is the machinery.

**Software** is the programs that run on top of the hardware to implement applications.
Hardware
RQ

How does a computer fit a large amount of data into its very small components?
How has this changed?

Eniac:

~19,000 pieces of data

http://www.computerhistory.org

My laptop:

~2,466,611,429,376 pieces of data
Moore’s Law

Computer speed and memory on a chip will double every 18 months

http://en.wikipedia.org/wiki/Moore's_law
Software
An **operating system** is a special kind of software that allows the other software to run. Probably most of you use Windows or Mac OS. In this course (especially in lab next week), we’ll also be teaching you how to use Linux, mostly the command line.
How does it all work?

We’ll cover how the hardware works later in the course.

For now, let’s concentrate on the software. Software works by programmers writing down algorithms in a special languages that the computer can understand.

An algorithm is a precise, systematic method for producing a specified result.
If programs are algorithms that have been specialized to a specific set of conditions and assumptions, how do program’s avoid and error or mistake as a result of a situation where the assumption does not indefinitely apply? Will such a situation occur? More specifically, what is an example of an assumption in a program?
RQ

To what extent can things in the physical world be made into algorithms?

I’d argue that many things that we do are algorithms. At the very least, many things that we teach are!
So that’s an algorithm. Now what?

Well, next we need to be able to turn it into something the computer will understand. Which means we need to be able to store our data.
A data structure describes relationships among data. There are many including:

- Network or graph data structures
- Hierarchical data structures
- Tabular data structures
Exercise: Data Types

write down two (universal) properties of each of the following data types; then compare with your neighbours’ properties.

• rectangles
• files (on a computer)
• mammals
• text (in a document)
Data Structures

**hierarchical structures** can be represented visually in many distinct ways

*Connecting with Computer Science*
www.ugrad.cs.ubc.ca/~cs101
network structures show relationships between pairs of data items
Directed vs Undirected Networks

- Recall description of networked data structure (page 4 module 1):
  - “describes general relationships between pairs of data items”
  - “links may be directed or undirected, depending on the type of the relationship”
  - symmetrical relationship $\rightarrow$ undirected
    asymmetrical relationship $\rightarrow$ directed
Directed vs Undirected Networks

Symmetrical relationships $\rightarrow$ undirected networks:
- people, animals: A is a relative of B
- people: A is a friend of B (e.g., on Facebook)
- countries: A trades with B

Examples for asymmetrical:
- people: A is an ancestor of B
- people: A likes B (people)
- folders/files: A contains B
- courses: A is a prerequisite for B
in **hierarchical structures**, data is organized at branch points and/or leaves of a “tree”
Matching Parentheses

- Parentheses provide a text-based way to describe hierarchical structure.
- What are pros and cons of text-based and graphical representations?

\[(A+(B+(C+D))))\]
**Convert: Text → Graphical**

**Step 1** (match the parentheses):
working from left to right,
- when you reach a “)”, match it with the last unmatched “(“ to its left
- use an arc to connect the pair of matched ( )’s so you remember which ones are matched
repeat until you get to the end

\[
( \ A \ + \ ( \ B \ + \ ( \ C \ + \ D \ ) \ ) \ ) \)
\]
Convert: Text $\Rightarrow$ Graphical

- result at end of Step 1
Matching Parentheses

**Step 2** (build the tree):

- put a node in the tree for each arc, and add links from each to the arcs and letters *directly nested* within it

```
  (   A   +   (   B   +   (   C   +   D   )   )   )
```
Matching Parentheses

- result at end of Step 2

( A + ( B + ( C + D ) ) )

Connecting with Computer Science
www.ugrad.cs.ubc.ca/~cs101
Matching Parentheses

\(( A + ( B + ( C + D ) ) )\)
Exercise: Graphical $\rightarrow$ Text
Create parentheses where there are edges

Convert this phylogenetic tree from graphical to text based representation.
So what does the computer do with these data structures?

• Different applications will deal with different data in different ways
• Sometimes they’ll treat the same data structure in a very different fashion…
RQ

Why would I want to give one file two different names?
RQ: why are there so few women in computer science

WOMEN IN SELECTED FIELDS
Bachelor's Degrees, 1966-2004

Percent of Bachelor's Degrees Awarded to Women, U.S.

Source: National Center for Education Statistics. Data for Academic Year 1999 were not available. Compiled by AIP Statistical Research Center.

Connecting with Computer Science
www.ugrad.cs.ubc.ca/~cs101
Learning goals revisited

• Understand what an algorithm is and why they matter
• Give an example of cases when one algorithm might perform better or worse than another algorithm
• Recognize examples of **data types** and illustrate how properties associated with familiar data types can influence the behaviour of computer applications which act on these data types
• Recognize examples of **data structures** and classify data structures as networked, hierarchical, and/or tabular when applicable
• Convert between small instances of graphical and text based representations of hierarchical structure