CS 105

Introduction to Scientific Computing

Topic #13 – Pseudocode, Algorithms and Problem Solving

Matt Burlick
Stevens Institute of Technology

TOPICS

1. What are algorithms?
2. How can we approach problem solving?

READING

- None

MOTIVATION

- Our programs/scripts are getting more complicated ☹
- Maybe it's clear how to trace existing scripts but how do we take a problem and make a script that may involve multiple
  - Branches
  - Loops
  - Etc..
- One approach is to re-word it to use the language of scientific computing
  - Analogous to solving word problems
- The process may involve
  - Coming up with an *algorithm* to solve the problem
  - Express this solution in generic *pseudocode*
  - Translate from pseudocode to the desired programming language
    - In our case, Matlab

WHAT'S AN ALGORITHM?

- An algorithm is a "**sequence** of **unambiguous** steps to complete a **task** in **finite** time"
- Sequence?
  - One after another
- Unambiguous
  - Clear, can't be confused
  - Should be very simple
- Task
  - The problem we're trying to solve
- Finite
  - The algorithms should eventually finish (even if it takes a reallllllly long time)

EVERY DAY ALGORITHMS

- How to go from the Path to Stevens?

COMPUTER ALGORITHMS

- Sorting
- Searching
- Shortest path in a network

CODE VS PSEUDOCODE

- There are so many programming languages out there!
    - C
    - C++
    - Java
    - Python
    - PHP
    - Matlab
    - Etc….
- Often we want to express an algorithm in a ***generic*** way so that anyone with programming experience can convert it into the language of their choice.
- We call a generic algorithm ***pseudocode*** and a formally structured algorithm for a particular programming language ***code***

PROBLEM SOLVING

- So how can we come up with an algorithm to solve a problem?
- Try re-phrasing it using the language of scientific computing
  - Unambiguous commands include:
    - Assigning a value
    - Doing a basic math operation
    - Branching based on conditions
    - For loops or while loops
  - To re-wording the problem to use things like

• "set x=…"	%assignment/computation
• **"If"**	%branching
• **"Foreach"**	**%for loop**
• **"While"**	**%while loop**
• **"Get"**	**%input**
• **"Display/Output"**	**%disp**

EXAMPLES I

- Sequential Search
  - Find value
- Statistics
  - Sum
  - Mean
  - Standard Deviation
  - Find min/max
- Sorting
  - Selection Sort
- Augmenting Vector/Matrix Data
  - Perform operation on each element

EXAMPLES II

- Keep asking user for input until they type a number
- Keep adding until user types a non-numerical value
- Computed average of numbers typed in by user until they type a non-numerical value