Lecture 1 - 1/16/2024 Last Updated: 1/10/2024

Course logistics - you can view all of these on the canvas homepage which every student should spend AT LEAST 5 MINUTES reading

Preliminary definitions:

- **Algorithm**: A well-ordered collection of unambiguous and effectively computable operations that when executed produces a result and halts in a finite amount of time.

- **Computer Science:** The study of algorithms including: 1. Their formal and mathematical properties 2. Their hardware realizations 3. Their linguistic realizations 4. Their applications

Your first Algorithm:

Linear Search (sequential search): Algorithm to determine whether a given name x appears in a list.

Input: A list of $n \geq 1$ names A[1], A[2], ... , A[n], and a given name x.

Output. The message "Sorry, x is not on the list" if x is not on the list. Otherwise, the message: "x occurs at position i on the list."

A *programming language* is a notation for specifying instructions that a computer can execute

- Every (programming) language has a syntax and a semantics
  - Syntax specifies how something is said (grammar)
  - Semantics specifies what it means
- Pseudocode is an informal programming language with English-like constructs modeled to look like statements in a Java-like language
- Anyone able to program in any computer language should understand how to read pseudocode instructions.
- When in doubt just write it out in English.

Lecture 2 - 1/18/2024

Linear search is quite inefficient when it comes to large chunks of data or even small chunks of data but the element you are looking for is towards the end of the collection of data. An improved searching method is known as **Binary Search** and it cuts the remaining data set in half after each iteration.

Binary Search:
Input: A SORTED list of $n \geq 1$ numbers A[1], A[2], ... , A[n],
and a target number x.

Output. The message "Sorry, x is not on the list" if x is not on the list. Otherwise, the message: "x occurs at position i on the list."

The key to Binary Search is starting in the middle of your list and removing half of the entries with each iteration, this is how we get the algorithmic efficiency for Binary Search which will be described in more detail later.

Try comparing run throughs of the previously mentioned algorithms on the following list A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] with x = 9