Week 12, Lecture 1

In today's lecture we will look at (a) finding the minimum and maximum of numbers in a list using a single pass method, and (b) two search algorithms — linear search and binary search.

Program 1 is one you have seen before (finding min or max of 3 numbers, last part of **Chapter 7** of the text). So please review this briefly.

Program 2 reads in data from a file and executes two separate loops, one to find the minimum number and one to find the maximum number. By now you know that the number of comparisons to find either the minimum or the maximum in a list of size n is O(n) because the algorithm has to touch all n number for each case.

Because we are using 2 separate loops, you could say that the work required to find both the minimum and then the maximum is O(2n).

Do you suppose we could cut down on some comparisons if we found both the minimum and the maximum in ONE loop (I.e., in one pass) ? The answer is yes.

In **file 3.1.py** (not a program) we explain the basic idea with an example, and in **Program 3.2** we have the Python code for the example. It's pretty simple, so please make

sure you read and understand it. We also explain how the work now becomes O(1.5n) instead of O(2n). It's not a huge reduction but the idea is interesting.

Program 4 explains the Linear Search using both Python's built-in index() method for lists, as well as our own search function.

Program 5 shows you how to do a binary search. The code is self-explanatory, but we have included a hand-written example (see pdf in folder) to make it a bit easier, for a small list. This is an important search method so be sure you understand it clearly. It is a simple and useful method with important application as a "divide-and-conquer" method. You will this idea a lot in computer algorithms.

Please read the first part of **Chapter 13** of the text for the linear search and the binary search algorithms.

Note: In last Wednesday's lecture folder I added (on Wednesday) a graphics folder containing a simple example of how to use two mouse function calls in graphics. I did this because I got a question on this after Monday's lecture.