CS 101 Lab 2

Searching *Paul Cao*

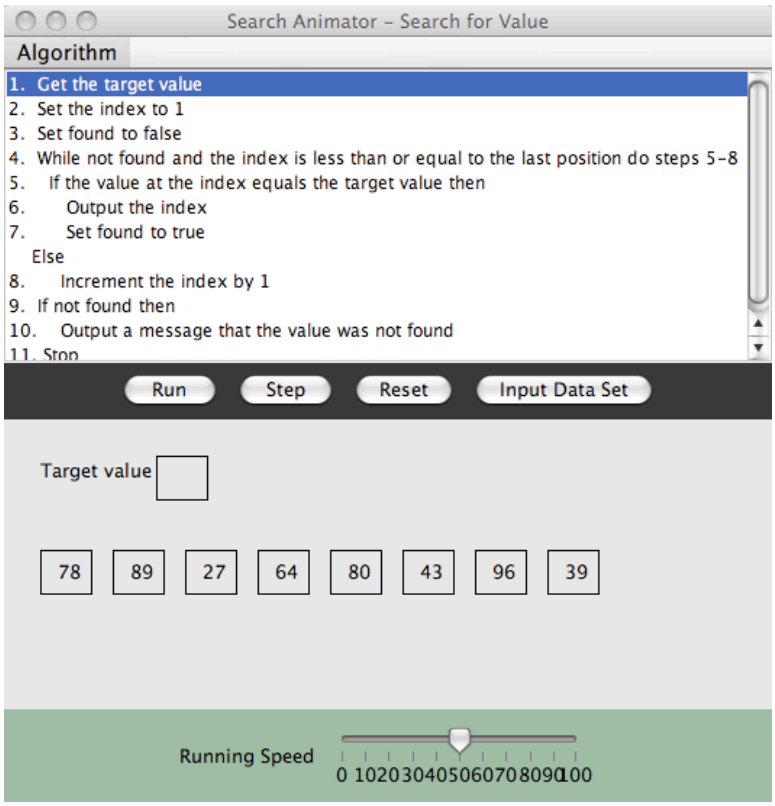
The lab report is due before class on Wednesday 2/9/2011. You need to write your answers for all the exercises in a word file. Submit your word file through the lab2 dropbox on angel.

Objectives

- Solidify your understanding of search for the largest value algorithm and sequential search with an algorithm animator.
- Use empirical data obtained via the animator to derive general mathematical relationships.
- Get a first glimpse at how computer scientists analyze algorithms to obtain measures of performance in terms of the size of the problem to be solved.

Background

The lab software should be on your desktop since you have downloaded it for lab 1. If you need to download it again, go to the lab folder on ANGEL. To begin the lab, click the **Search Animator** button on your lab software menu. At this time, a new window will appear. Note the following components



- An **Algorithm** menu; this menu lets you select the algorithm to run (either search for a given value or search for the maximum value). You can make a selection on this menu after you click the **Reset** button.
- The text of the current algorithm (search for a given value when the window is opened)
- A set of four command buttons (**Run, Step, Reset,** and **Input Data Set**)
- A graphic image of an array of values and a box waiting to be filled (with the user's target value for search for a given value or with the current maximum value for the search for the maximum value) .
- A slider bar to control the running speed. The higher the number is, the faster the simulation will run.

Before you start the simulation, you can click on the **Reset** button, and a new set of data will be generated. If for some reason you would like to see how the algorithm performs for some specific data set, you can click on the **Input Data Set** button. You will be presented with an array of cells into which you can enter the specific values you wish to try. Notice that the first step of the algorithm is shaded. As the execution progresses, the next step to be executed will be shaded. This will enable you to keep track of exactly where the algorithm is in its execution and correlate what is happening to the data on the screen

with the step of the algorithm being executed. If you click on the **Run** button, the animator executes the entire algorithm, pausing only for input and output. Clicking on the **Step** button causes just a single step of the algorithm to be executed or pauses execution during a run. Clicking the **Run** button during stepping resumes the execution as a run. We will primarily use the step mode of operation in this lab.

You may notice that the pseudo-code differs slightly from what we learned in the lecture. This is fine since pseudo-code isn't intended to be rigid and exact, but rather provide the constructs needed to convey in a fairly precise way how the algorithm performs its task.

Exercise 2.1
Select the **Search for Largest** option from the **Algorithm** menu. Use the **Step** button to execute the algorithm two or three times until you understand exactly how the algorithm works and how the animation corresponds to the steps in the pseudo-code window. Remember to click **Reset** button when you are ready to start over with a new set of data. When clicking on the **Step** button, you need to hold the **Step** button a little bit longer than regular single clicks to allow calculations to be done.

As you will observe, the **Location** pointer changes its position during the execution of the algorithm. Can you answer the following questions?
Q1: What would be the smallest number of positions that Location pointer would possibly take on? Describe the specific conditions under which this would happen. Give a sample of data for this condition.

Q2: What would be the largest number of positions that Location pointer would possibly take on? Describe the specific conditions under which this would happen. Give a sample of data for this condition.

Exercise 2.2
Select the **Search for Value** option from the **Algorithm** menu. Use the **Step** button to execute the algorithm two or three times until you understand exactly how the algorithm works and how the animation corresponds to the steps in the pseudo-code window. Remember to click **Reset** button when you are ready to start over with a new set of data. Like before, when clicking on the **Step** button, you need to hold the **Step** button a little bit longer than regular single clicks to allow calculations to be done.

Press the **Reset** button to generate a new list of data. Choose the first element in the list as the target and step through the algorithm. As you step, keep a careful tally of exactly how many times each step is executed. Enter the data into the first row of the table into the Worksheet (**include the worksheet electronically in your lab report**). Compute the total number of steps executed and enter this into the table. Now reset the animator, use the second data element as the target, and enter the results into the second row of the table. Continue to do this until a distinct pattern becomes apparent.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10
1st element										
2nd element										
3rd element										
4th element										
5th element										
6th element										
7th element										
8th element										

Q3: What is the pattern that you found for the number of executions for each step if the target is the k^{th} item in the list? Hint: some steps are executed in a constant time no matter what k is. But some steps' executions are based on k. Your answer to those steps should be related to k, such as k+1, k-1, etc.

Step 1:
Step 2:
Step 3:
Step 4:
Step 5:
Step 6:
Step 7:
Step 8:
Step 9:
Step 10:

Q4 (extra credit): Can you find the pattern for the total number of steps for the entire algorithm if the target is the k^{th} item in the list?

When you are done with the lab, log out the computer.