

Lab 12 – Recursion and Binary Search

CPSC 1020

Due: Thursday, April 20, 2017 Midnight

INTRODUCTION:

This lab is designed to give you practice working with recursion and the binary search algorithm.

Please watch the following videos before attending lab:

<https://www.youtube.com/watch?v=VrrnjYgDBEk>

<https://www.youtube.com/watch?v=beqqGldabrE>

<https://www.youtube.com/watch?v=5xIPT1FRcA>

https://www.youtube.com/watch?v=MZaf_9IZCrc

LAB OBJECTIVES:

In addition to practicing concepts used in previous assignments and labs, upon completion of this lab, you should be able to:

- Write a recursive quick sort function
- Search for an element in a sorted array using a recursive binary search

SUBMISSION INSTRUCTIONS:

Submit your files to handin (

<http://handin.cs.clemson.edu>)

by the due date announced in class as a

compressed tar.gz file named username-lab12.tar.gz. You will need to submit driver.cpp, functions.cpp, functions.h, and makefile (small “m”)

.

LAB INSTRUCTIONS:

Download the necessary starting files from canvas.

For this assignment, you are going to write several functions, which are implemented in the **functions.cpp** file.

void print_array(int array[], int array_size);

This function basically prints out the content of the array.

Lab 12 – Recursion and Binary Search

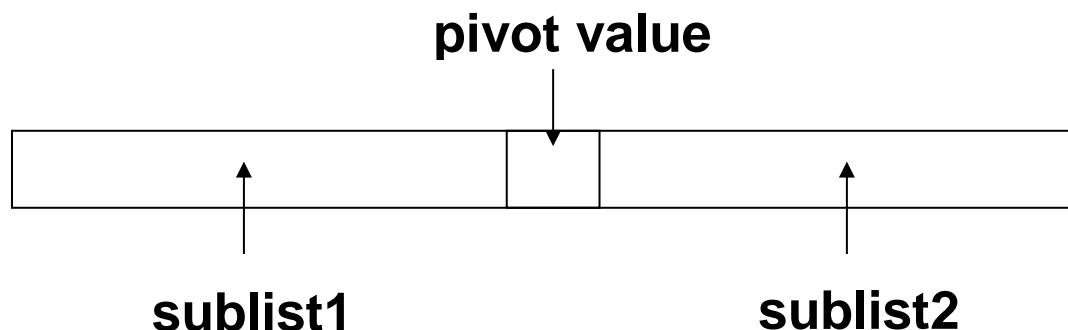
CPSC 1020

Due: Thursday, April 20, 2017 Midnight

int partition(int array[], int start, int end);

This function will perform the following steps:

- Select a pivot element from array
- Rearrange array into sublist1, the pivot element, and sublist2 so that the pivot element is at position **p** and sublist1 and sublist2 are, respectively, **array[start--p-1]** and **array[p+1 – end]**



- Return the position **p** of the pivot

This function will require you to swap some values. In functions.cpp, I have included the **algorithms** library which will allow you to use the C++ provided swap function. However, you can write your own if you wish. **MAKE SURE YOU WATCH THE QUICKSORT VIDEO PROVIDED.**

void quickSort(int array[], int start, int end);

This is a recursive function that will first call the partition function to determine the pivot point. If the value of start is less than the value of end. It will then call itself twice.

- For sublist1 (see above image)
- For sublist2 (see above image)

int binarySearch(const int array[], int first, int last, int value);

This is a recursive version of a binary search function. It returns the subscript of where the value is found or -1 if the value is not found

array is what you will be searching

first is the subscript for the first element in the search range

last is the subscript for the last element in the search range

value is the value to be searched

This algorithm can be described in the following way:

Lab 12 – Recursion and Binary Search

CPSC 1020

Due: Thursday, April 20, 2017 Midnight

The base case is met when you have searched the entire array and did not find the value you were looking for.

If the base case is not met, then:

1. Determine the middle element of the array.
2. If the middle element is the value you are looking for then return the subscript of the middle element.
3. If the value for the middle element is less than the value you are looking for then recursively call `binarySearch` on the elements to the right of the middle.
4. Otherwise recursively call `binarySearch` on the elements to left of the middle. Remember `binarySearch` has a return.

driver.cpp

The steps needed in the driver:

1. Open a file to read in the data for the array.
2. You will first read in the number of elements to be stored in the array.
3. Next read in the values for the array.
4. Close the file pointer after you have read in all values.
5. Print the elements of the array.
6. Sort the values in the array using the quicksort function.
7. Print the elements of the sorted array.
8. To test the search algorithm, ask the user to enter a number to search for.
9. Call the `binarySearch` function.
 - a. If the function did not find the value print a message telling the user the number was not found
 - b. If the value was found print a message telling the user, the value was found and what subscript contained the value.

makefile

You should also provide a makefile (lowercase m). Your makefile should create a target that will compile your program and a target that will remove the executable. To compile I should be able to type `make`. To remove the executable, I should be able to type `make clean`.

The makefile should also provide a target called `run` that will run the compiled program with the `input1.txt` file and then again with the `input2.txt` input file.

Lab 12 – Recursion and Binary Search

CPSC 1020

Due: Thursday, April 20, 2017 Midnight

FORMATTING:

You will need to add a header to each of your files like the following:

```
/*****
```

```
*Your name
```

```
*CPSC 1021 your section, F16
```

```
*Lab 12
```

```
*Your username
```

```
*****/
```

Your program should compile without warnings and with no errors.

Your code should be well documented with comments.

You should use proper and consistent indentation.

You **MUST** test your program on one of the School of Computing servers prior to submitting.

Your program will be tested on the SOC lab machines.