

COMP108 Algorithmic Foundations

Tutorial 4

w/c 27th February 2017

Tutorial participation contributes to 5% of overall marks. For this tutorial, make sure you have scanned your ID card.

You can refer to the lecture notes (“Searching” - Slides #6 & #11 and “Time Complexity” - Slide #14) for the pseudo codes.

<http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201617/notes.html>

1. Download two java files SearchApp.java and Search.java from the tutorial page the module website. <http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201617/tutorial.html>

(Use right mouse click to save the file.)

Mind that the lecture notes assume numbers are stored in $a[1], a[2], \dots, a[n]$ but the programs assume numbers are stored in $data[0], data[1], \dots, data[count-1]$.

Also note that the size of the array can be much larger than the number of values that the user input. The variable *count* stores how many values the user has input.

When you test your program, you should try to find each individual number one by one to make sure that a number can be found regardless of its position in the array.

- (a) Compile and run the program; then enter the following numbers **50, 40, 30, 20, 10**, one per line, followed by **-1** to terminate the input. Try the options to find the minimum number, sequential search for any number or binary search for any number. Note that these functions are NOT working yet. Other test cases:

- i. **10, 30, 20, 40, 50**

- ii. **50, 30, -10, 40, 20**

- iii. **-10, -20, -30, -15, -5**

- (b) Fill in the program Search.java the method **sfind()** to find if the input number exists in the array or not using sequential search. The array to be searched is `data[]`.

Remember to read the comments in the method.

When you test your program, you should try to find each individual number one by one to make sure that a number can be found regardless of its position in the array.

- (c) Fill in the program Search.java the method **findMin()** to find the minimum number in the array and test if it works. The array to be searched is `data[]`.

Remember to read the comments in the method.

- (d) Fill in the program Search.java the method **bfnd()** to find if the input number exists in the array or not using binary search. The array to be searched is `data2[]`, which contains the numbers in `data[]` in an ascending order (from smallest to largest).

Remember to read the comments in the method.

When you test your program, you should try to find each individual number one by one to make sure that a number can be found regardless of its position in the array.