HOMEWORK 3

100 points

DUE DATE: September 29th 11:59pm.

Warning: For any homework assignment that contains a programming segment please read the following very carefully.

Your code must compile and run on Black server. If your code does not work on Black server as submitted the grade for that problem is 0 regardless of its quick fix(es). Always test your code on Black, even if it is incomplete, make sure to get your code to compile and run on our Servers.

This homework contains 4 problems.

With your requirements document you are given:

1. An incomplete SearchSort2.h
2. main.cpp
3. makefile And
4. inputfile

Please do not modify main unless it is specified in the problem

Problem 1

Referring back to Homework 2, Problem 3 → Linear Search, observe that if the sequence A is sorted, we can check the midpoint of the sequence against *v (search value)* and eliminate half of the sequence from further consideration. This algorithm is known as Binary Search. Binary search repeats this procedure, cutting the size of the remaining portion of the sequence each time.

a) Write the **pseudocode for Binary Search using iterative version** (using a loop). As in Homework 2 you can use paper or a text editor for this part, make sure to address your name and the problem number, and convert your document to PDF prior to submission.

b) Using the incomplete template (SearchSort2.h) provided for you convert your pseudocode to C++ code for this iterative Binary Search. Do not modify the method signature. The parameter list stays the same.

```
Comparable binarySearch(vector<Comparable> &items, const int key)
```

c) Write the **pseudocode for Binary Search using the recursive version**. As in Homework 2 you can use paper or a text editor for this part, make sure to address your name and the problem number, and convert your document to PDF prior to submission.

d) Using the template provided for you convert your pseudocode to C++ code for this write for the recursive version of Binary Search.

Do not modify the method signature shown below, your method signature should be like that. The parameter list stays the same. Hint: Feel free to add a recursion helper method **that actually does the recursive call** and you can define this method with more parameters that you may need and call this helper method from **recBinSearch**. This concept is known as **encapsulation**. Encapsulation is heavily used with recursive methods in any language uses OOP.

The method that calls the **helper** method (this helper method is usually the one with the recursion) is referred as the driver. The driver calls the actual method that contains the recursion. This is done so that main method or any other method from any other class calls the **recBinSearch** with minimal possible parameters without cluttering the call with list of parameters. The goal is to hide the details from main or any other method (a.k.a encapsulation) that needs to call your Binary Search.

```
Comparable recBinSearch(vector<Comparable> &items, const int key)
```

Remember all codes written in .h files must compile and run with our main method on Black Server.

Problem 2

You wrote the Binary Search using recursion in Problem 1.
In this problem, write the equation or inequality that the describes your Recursive Binary Search function in terms of its value on smaller inputs.
For example:

$$T(n) = \begin{cases} \Theta(1) & if \quad n = 1. \\ 2T(n/2) & if \quad n > 1. \end{cases}$$

where the first line in curly braces is the base case is when n=1 and second line simply demonstrates the recursive behavior.
Using the same approach demonstrated as an example above, write your T(n) for your Recursive Binary Search function.

Problem 3

After completing Problem 2, using the Master Theorem solve

$T(n)$, make sure to specify which case it satisfies and display its growth function based on this case. Please note (if you need the following) that k=1 and ε(epsilon)=0.

Describe in very short sentences: Binary Search recursive algorithm's **best case, and worst case.**

Problem 4

Using the Master Theorem solve:

$$T(n) = \{2T(n/2) \ + \ n^3\} \quad where \ \ \varepsilon = 2 \ , \kappa \geq 0$$

make sure to specify which case it satisfies and display its growth function based on this case.

Homework 3 Deliverables:

The following files must be submitted via Handin no later than September 22nd 2016 by 11:59pm

1. **SearchSort2**.h
2. **Problem1 (a,c) and Problem 2, Problem 3, Problem 4 in PDF format.**

References:

1) Getting Started, Introduction to Algorithms, The MIT Press, Cormen, Liesorson, Rivest, Stein
2) Wikipedia: Sorting Algorithms