

University of Newcastle
Discipline of Computing and Information Technology
Semester 2, 2018 - SENG1120/6120

Assignment 3

Due using the Blackboard Assignment submission facility:

11:59PM – November 9th, 2018

NOTE: *The important information about submission and code specifics at the end of this assignment specification.*

INTRODUCTION

In lectures, we have discussed the benefits of using binary search trees and hash tables to store information. In this assignment you will implement both and compare their performances in terms of speed of access.

ASSIGNMENT TASK

You are required to create a binary search tree and a hash table data structures to store integers. Both structures should have functions to add, search and remove elements, and **MUST** be class templates. The binary search tree class must be called `BSTree` and will use as nodes instances of `BTNode`. The hash table class must be named `HTable`.

You will be provided a demo file and your classes need to interface with it. The binary search tree contents must be printed using an **inorder traversal**. The hash table class must store the numbers in an array of integers with size 150 and the contents can be printed from position 0 to n-1, but only for those positions that contain a valid entry. The hash function used must be:

```
int hashfun(int value)
{
    return value%150;
}
```

SUBMISSION

Make sure your code works with the files supplied, and DO NOT change them. For marking, we will add the main file to the project and compile it using the makefile, together with your own files. If it does not compile or run, your mark will be zero.

Your submission should be made using the Assignments section of the course Blackboard site. **Incorrectly submitted assignments will not be marked.** You should provide all your files. Also, if necessary, provide a `readme.txt` file containing any instructions for the marker. Each program file should have a proper header section including your name, course and student number, and your code should be properly documented.

Remember that your code should compile and run correctly using Cygwin. There should be no segmentation faults or memory leaks during or after the execution of the program.

Compress all your files, including the cover sheet, into a single .zip file and submit it in by clicking in a link that I will create in the Assignments section on Blackboard especially for that. Late submissions are subject to the rules specified in the Course Outline. Finally, a completed Assignment Cover Sheet should accompany your submission.

This assignment is worth 15 marks of your final result for the course.

```
/home/SENG1120/Ass3
$
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
Alexandre@ces249-339952s /home/SENG1120/Ass3
$ ./assignment3.exe
=====
BINARY SEARCH TREE
Initial tree: 5650 23418 34465 56534 97567 123454 345169 565471 678760 787626 867570 879840 1456769 3456462 5465443
Final tree : 5650 23418 34465 56534 97567 123454 345169 565471 678760 787626 867570 879840 1456769 3456462 5465443
Time elapsed: 0.234 seconds
Time per ins/del operation: 0.167143 milliseconds.
=====
HASH TABLE
Initial hash table: 123454 678760 3456462 23418 345169 5465443 97567 879840 5650 34465 1456769 867570 565471 787626 56534
Final hash table : 123454 678760 3456462 23418 345169 5465443 97567 879840 5650 34465 1456769 867570 565471 787626 56534
Time elapsed: 0.094 seconds
Time per ins/del operation: 0.0671429 milliseconds.
The program has finished.
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
Alexandre@ces249-339952s /home/SENG1120/Ass3
$ ./assignment3.exe
=====
BINARY SEARCH TREE
Initial tree: 5650 23418 34465 56534 97567 123454 345169 565471 678760 787626 867570
879840 1456769 3456462 5465443
Final tree : 5650 23418 34465 56534 97567 123454 345169 565471 678760 787626 867570
879840 1456769 3456462 5465443
Time elapsed: 0.234 seconds
Time per ins/del operation: 0.167143 milliseconds.
=====
HASH TABLE
Initial hash table: 123454 678760 3456462 23418 345169 5465443 97567 879840 5650
34465 1456769 867570 565471 787626 56534
Final hash table : 123454 678760 3456462 23418 345169 5465443 97567 879840 5650
34465 1456769 867570 565471 787626 56534
Time elapsed: 0.094 seconds
Time per ins/del operation: 0.0671429 milliseconds.
The program has finished.
Alexandre@ces249-339952s /home/SENG1120/Ass3
$
```

Obs: the computational time for each data structure will depend on the computer, but it is expected that the hash table will be faster than the binary search tree.