# EECS 560 Midterm Exam Section 1

## Note: This exam is closed-book and closed-notes

**Name: Chase Odgers** **KU ID: 2868660**

**(1) Describe the advantages of passing rvalue reference as parameter (10pts).**

Passing parameters can be resource intensive when passing as value, all constants get copied and this can present issue when dealing with large parameters, rvalue and lvalue allow for the argument to be passed by reference avoiding this copy, with lvalue however the constants are able to be modified by the function they are being passed to as they are not "read only", passing by rvalue allows for the the parameter to be passed as a "const" effectively marking it "read only" and allowing both the resource advantage of passing by reference while avoiding overwriting any variables within the parameter being passed.

**(2) What is the worst-case scenario time complexity of searching for a specific element from a binary search tree? (10pts)**

The worst-case scenario time complexity of a BST is **O(n).**

**(3) Explain why rehashing is required when table doubling/halving happens. (10pts)**

Hash tables are given a fixed size, when the number of entries in a hash table becomes too large it is necessary to grow the table to avoid overflow, likewise when it becomes too small we need to shrink the table, these processes are doubling/halving respectively. This resizing demands the previous entries be replaced in the new table and to correctly disperse them within the new table they must be re-hashed, or ran through the "hashing" operation again, where the "hashing" operation dictates their location in the table.

**(4) What are "Big Five" defined in the context of C++ object interface? (10pts)**

In C++ there are certain methods that most objects require, the "big five". These methods are: **destructor**, **copy constructor**, **move constructor**, **copy assignment, move assignment**.

**(5) In a 64-bit machine, how much memory space would a binary search tree Node designed for searching integers occupy? (10pts)**

Binary search tree nodes contain their data and a pointer to their left node and their right node. A pointer on a 64-bit machine is 64-bits. 128-bits + int(size), if int is 8-bits(assuming it takes 1 memory address) the answer would be 192 bits or 24 bytes. I am unsure on the size of an int.