1. Rank a set of Big-O notations
   1. O(n log n)
   2. O(n2)
   3. O(log n)
   4. O(n)
   5. O(n!)
   6. O(1)
   7. O(2n)
2. What is the computational complexity of the following function? n3+30nlogn + 12000
3. Identify the “critical section” in calculating computational (asymptotic) complexity.
4. Convert a binary tree to an array using the 2i+1/2i+2 rule.
5. Convert an existing heap into a heap that results from inserting or deleting a node.
6. Sets and Maps: Write code that adds a mapping to a key-value mapping; write code that displays each value and key
7. Identify valid hash-codes for a class (returns an int)
8. Does a given class satisfy the hash-code contract?
9. Identify the fastest (in Big-O) of a set of sort algorithms
10. Identify the fastest (in Big-O) of a set of search algorithms
11. Implement an unchecked and a checked exception
12. Implement a method that handles an exception
13. Modify a class to make it generic
14. Insert/retrieve a node from a binary search tree
15. Write a recursive method.
16. Instantiate a tree given a generic class, get an iterator (already defined), and use it
17. Implement a for-each loop on the same tree