CS 1332 Exam 3 - SECTION A

Spring Semester: April 18, 2016

Name (print clearly including your first and last name): _____

Signature: _____

GT account username (gtg, gth, msmith3, etc): _____

CIRCLE your section: A1 A2 A3 A4 A5 or A6

- Signing and/or taking this exam signifies you are aware of and in accordance with the **Academic Honor Code of Georgia Tech** and the **Georgia Tech Code of Conduct**.
- Notes, books, calculators, cell phones, laptops, headphones, etc. are not allowed.
- Extra paper is not allowed. If you have exhausted all space on this test, talk with your instructor.
- Pens/pencils and erasers are allowed.
- All code must be in Java.
- Efficiency matters. For example, if you code something that uses O(n) time or worse when there is an obvious way to do it in O(1) time, your solution may lose credit. If your code traverses the data 5 times when once is sufficient, then this also is considered poor efficiency even though both are O(n).
- Style standards such as (but not limited to) use of good variable names and proper indentation is always required. (Don't fret too much if your paper gets messy, use arrows or whatever it takes to make your answer clear when necessary.)
- Comments are not required unless a question explicitly asks for them.

Name: _____

[5] 1. Explain what it means for a sorting algorithm to be stable.

[5] 2. What size alphabet is the Boyer-Moore text processing algorithm MOST useful with? Why?

 Points on this page: 10 points.

Name: _____

3. Fill in the blank. Write the Big-O using proper notation. (For example, do not write just n, but rather O(n).) In all cases, n refers to the number of data items in the data structure or characters in the text. m is the number of characters in the pattern. k is the largest number of digits of an integer in the array. V is the number of vertices in the graph, while E is the number of edges in the graph.

[2] (a) _____ What is the worst case big O of MSD radix sort?

[2] (b) _____ What is the big O of quick sort when a bad pivot is picked at every iteration?

[2] (c) _____ What is the big O of insertion sort when the list is already sorted?

[2] (d) _____ What is the worst case big O of breath first search?

[2] (e) _____ What is the worst case big O of updating the text hash in Rabin Karp?

[5] 4. (a) Which sorting algorithm does not require any comparisons between the data elements?

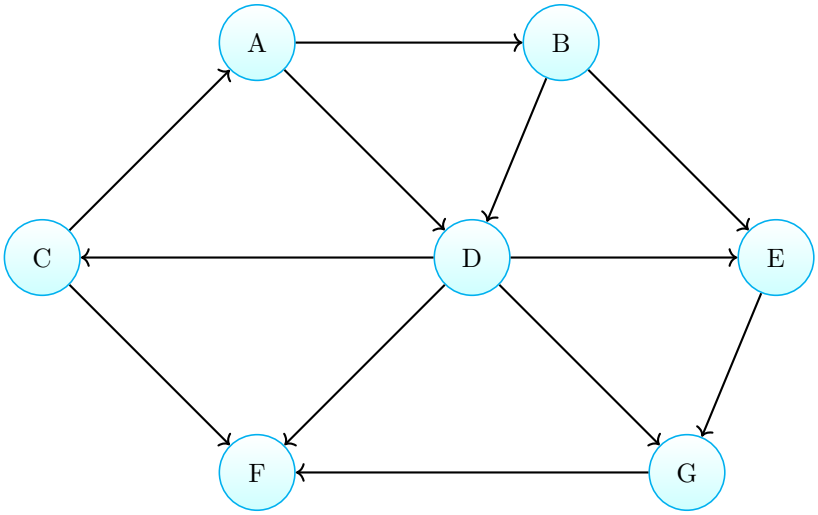
A. Radix Sort
B. Bubble sort
C. Quick sort
D. Merge sort

[5] (b) Which graph search algorithm can be computed using a Stack?

A. Brute force search
B. Breadth first search
C. Depth first search
D. Dijkstra's algorithm

Name: _____

5. Use the following graph for this question.



[10] (a) Make an adjacency list of the graph above.

[10] (b) Perform a BFS traversal of the graph above, starting at vertex A. In case of ties, visit the vertices in alphabetical order.

 Points on this page: 20 points.

Name: _____

[15] 6. Given the text "abcbcabbacbcabbca" and the pattern "bcabbca", and the failure table based on the pattern, use KMP to find the first instance of the pattern in the text. Use the following table to show your work. Each time you compare characters, circle the character in the pattern.

Failure table:

b	c	a	b	b	c	a
0	0	0	1	1	2	3

a	b	c	b	c	a	b	b	a	c	b	c	a	b	b	c	a
b	c	a	b	b	c	a										

How many comparisions did you make?

Points on this page: 15 points.

Name: _____

[10] 7. Given the below list of numbers, perform one iteration of cocktail shaker sort.

14 9 42 29 6 34 22 27

Points on this page: 10 points.

Name: _____

[25] 8. In the space below, implement the Rabin-Karp algorithm.

You may assume that `generateHash` and `updateHash` are already implemented and work as documented. Arguments to `rabinKarp` will never be `null`. The pattern will never be empty. **Do not** modify the code that is given (which includes not modifying the header that is provided for the method you are writing). **Do not** assume that any other methods exist. **Do not** add any other methods, private or otherwise.

```
/**
 * Hashes the first length characters of str.
 *
 * @param str the string to hash
 * @param length the number of characters to hash
 * @returns the integer hash
 * @throws java.lang.StringIndexOutOfBoundsException if length > str.length()
 */
private int generateHash(CharSequence str, int length) {
    // implementation omitted for brevity
    // you may assume that this method works as documented above
}

/**
 * Updates oldHash by removing oldChar and adding newChar.
 *
 * @param oldHash the existing hash
 * @param oldChar the character to remove from the hash
 * @param newChar the character to add to the hash
 * @param length the length of the hashed string
 * @returns the updated hash
 */
private int updateHash(int oldHash, char oldChar, char newChar, int length) {
    // implementation omitted for brevity
    // you may assume that this method works as documented above
}

/**
 * Finds the first occurrence of pattern in text using the Rabin-Karp algorithm.
 *
 * @param pattern the pattern to locate
 * @param text the text in which to search for pattern
 * @returns the index of the first occurrence of pattern in text, or -1 if
 *          no match exists
 */
public int rabinKarp(CharSequence pattern, CharSequence text) {









}
```

 Points on this page: 25 points.

The text on this page was shaken off by Taylor Swift. Do not remove this page.

 Points on this page: 0 points.

The text on this page was shaken off by Taylor Swift. Do not remove this page.