CS 1501: Algorithm Implementation
Fall 2019

Practice Midterm Examination

October 2019 Suggested: 60 minutes

Name: _____ **Username (abc123):** _____

Instructions:

- This is a closed-book, closed-notes practice exam. It is recommended that you take it without using any outside resources, print or electronic.

Problem 1: Carefully read each prompt and select the best answer for each question.

i. Which of the following is a stable sorting algorithm?

(a) LZW
(b) Radix sort
(c) Heapsort
(d) None of the above

ii. Which of the following is a property shared by all of the tries we discussed?

(a) Tied values always maintain their relative ordering
(b) Worst case height is guaranteed to be $O(\log n)$
(c) Keys are implicitly stored as paths from root to leaf
(d) Nodes are tuned to be 1 disk block in size

iii. Which of the following cannot be used to implement a symbol table?

(a) Rabin-Karp
(b) A B-tree
(c) A binary search tree
(d) All of the above can be used to implement a symbol table

iv. Assume that $f(x)$ is $O(x^2)$ and $f(x)$ is $\Omega(\log(x))$. Which of the following must be true?

(a) $f(x)$ is $\Theta(x)$
(b) $f(x)$ is $O(x^3)$
(c) $f(x)$ is $\Omega(x^3)$
(d) None of the above

v. Which of the following is a property of a good hash function?

(a) Utilizes most of the key
(b) Produces a normal distribution of hash values
(c) Produces the same output for every input
(d) Exploits differences between keys

Problem 2: The following code is used in the book's implementation of LSD Radix sort. Fill in the blanks so that the code works properly.

```
public static void sort(String[] a, int w) {
    int n = a.length;
    int R = 256;   // extend ASCII alphabet size
    String[] aux = new String[n];

    for (int d = w-1; d >= 0; d--) {
        // sort by key-indexed counting on dth character

        // compute frequency counts




        int[] count = ____________________________________________________;
        for (int i = 0; i < n; i++) {




            ____________________________________________________________;
        }
        // compute cumulates
        for (int r = 0; r < R; r++) {




            ____________________________________________________________;
        }
        // move data
        for (int i = 0; i < n; i++) {




            ____________________________________________________________;
        }
        // copy back
        for (int i = 0; i < n; i++) {
            a[i] = aux[i];
        }
    }
}
```

Problem 3: Consider using Huffman encoding to compress the following string.

`DONTTELLMEILLTELLYOU`

Show the following:

- The Huffman tree
- The codeword/character pairs

State any assumptions that you make.

Problem 4: Use separate chaining to insert the following numeric keys into a hash table and show the resulting data structure. Insert the keys in the order shown (from left to right). Use the hash function provided.

$$h(x) = x \bmod 7$$

Values to insert: 14, 5, 3, 7, 15, 17, 6, 23

Problem 5: Consider the alphabet {A, B, E, N}.

Construct a de la Briandais trie over this alphabet containing the following words:

BEE, NAB, BEAN, BAN, BEEN, BAE

Be sure to make the details clear, such as how complete keys are differentiated from prefixes, and how references are organized in the node.

Problem 6: Consider the alphabet {R, S, T, U}, represented by the 2-bit block codes $\{0, 1, 2, 3\}$, respectively. Expand the LZW-compressed string represented by the following sequence of (4-bit) codewords:

```
0 3 1 2 2 3 2 6 4 11 5 2
```

For each codeword processed, show the output character(s) and the new codeword added to the codebook.

Problem 7: Consider the following B-tree.

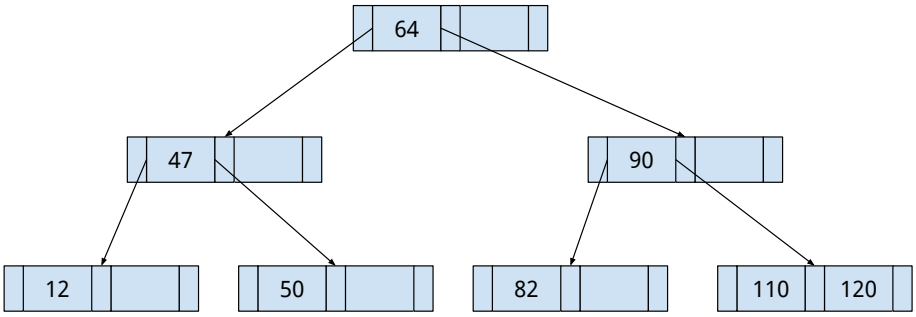

Figure 1: An example B-tree

i. Show the B-tree that would result from inserting 115 into the B-tree shown in Figure 1.

ii. Show the B-tree that would result from removing 12 from the B-tree shown in Figure 1.

iii. What is the order of the B-tree shown in Figure 1?