CSC 440 - Assignment 7

# Instructions

First, make a copy of this document (click on File → Make a copy). You will be editing your own copy of the assignment.

When you finish the assignment, you must download your copy and submit the downloaded file into Moodle (click on File → Download → File type; please use .doc, .docx, .pdf, or .rtf formats for your submissions.)

# Problems

1. (15 pts) Sort the following arrays using *Sorting by Counting* and by *Distribution Counting*. For *Sorting by Counting*, show the state of the *count* array at each step of the process. The final state of the *count* array shows the indices of each element in the input array when they are in sorted order, so arriving at the final state of the *count* array is sufficient to answer the question. For *Distribution Counting*, simply show the array at each step while filling it in. In other words, the first row of this answer should show an array of the proper size that is empty except for one element; the second step will show an array that is empty except for two elements, etc.

[4, 6, 1, 9, 2, 7, 3] (for sorting by counting)

[2, 6, 2, 2, 1, 6, 7, 5, 2, 1] (for distribution counting)

1. (15 pts) Use Horspool’s to find the pattern **CCAC** in the following string. Fill in the provided shift table, and then use the large table provided to show your steps, adding a new row for each starting position of the pattern.

**Text string:** ACAAGCACGCAGCCACTAC

| c | t(c) |
| --- | --- |
| A |  |
| C |  |
| Everything else |  |

| A | C | A | A | G | C | A | C | G | C | A | G | C | C | A | C | T | A | C |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | C | A | C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. (15 pts) In your own words, describe the difference between *Open Hashing* and *Closed Hashing*. Name at least one benefit and one drawback of each approach.
2. (15 pts) What is the maximum possible height *h* of a B-Tree of order 8 with 1000 nodes?
3. (**Extra Credit** 5 pts) Use the Boyer-Moore algorithm to complete the substring search from Problem 1. Fill in the provided shift tables, and then use the large table provided to show your steps, adding a new row for each starting position of the pattern. In the leftmost column, record which of the two rules you use in that step to determine the next shift (simply type “B” for Bad Character, and “G” for Good Suffix). The first row has been filled in already.

**Pattern:** CCAC

**Text string:** ACAAGCACGCAGCCACTAC

| k | pattern | d2 |
| --- | --- | --- |
| 1 | CCAC |  |
| 2 | CCAC |  |
| 3 | CCAC |  |

| c | t(c) |
| --- | --- |
| A |  |
| C |  |
| Everything else |  |

|  | A | C | A | A | G | C | A | C | G | C | A | G | C | C | A | C | T | A | C |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **B** | C | C | A | C |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |