CSCI 3104 Fall 2021
Instructors: Profs. Grochow and Waggoner

Quiz- Standard 18

Due Date ..... TODO
Name ..... **John Blackburn**
Student ID **Jobl2177**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students**. **Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.

- My submission is in my own words and reflects my understanding of the material.
- I have not collaborated with any other person.
- I have not posted to external services including, but not limited to Chegg, Discord, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (John Blackburn). □

3 Standard 18: Divide and Conquer.

Problem 2. Given an array $A[1, ..., n]$, we say it contains a duplicate if there are two distinct indices $i \neq j$ such that $A[i] = A[j]$. Consider the following divide and conquer algorithm for counting the number of duplicates.

```
countDuplicates (A[1, ..., n], integer p, integer q):
    if length(A) == 2 {
        if A[0] == A[1] {
            return 1
        }
        else {
            return 0
        }
    }
    if q > p {
        r = floor ((p+q)/2)
        L = countDuplicates (A, p, r)
        R = countDuplicates (A, r+1, q)
        return L + R
    }
    else {
        return 0
    }
```

Will the above algorithm return the correct number of duplicates? **Explain and justify** your answer by computing `countDuplicates(A, 1, n)` and showing what the algorithm does at each step. If the algorithm does not correctly count duplicates, give an example to illustrate that the algorithm fails.

Answer. No, the algo will not compute the correct amount of duplicates. I'm assuming that n denotes the last element within the array which can be any number depending on the array input. The algorithm will never compare anything other than the first two elements in the array. The array is never split up into smaller arrays, so the base case comparison will never take place unless the original array input is of length 2. Take for example the input cD(A[1,0,1], 1, 1). The algo skips the first if statement because the length$(A) = 3$. It then skips the next if statement because $q = 1$ and $p = 1$. It then goes to the else statement and returns 0, telling us that the array has 0 duplicates when in fact $A[0] = A[2]$, so the correct answer is 1. So, overall the algorithm fails to provide the correct amount of duplicates.

□