CSCI 3104 Spring 2023
Instructors: Chandra Kanth Nagesh and Prof. Ryan Layer

Quiz 18 Standard 18 – (Divide & Conquer) Basics and CounterExamples

Due Date .... TODO
Name .... **Blake Raphael**
Student ID **109752312**
Quiz Code (enter in Canvas to get access to the LaTeX template) .... **ASZZZ**

Contents

Instructions

- You may either type your work using this template, or you may handwrite your work and embed it as an image in this template. **If you choose to handwrite your work, the image must be legible, and oriented so that we do not have to rotate our screens to grade your work.** We have included some helpful LaTeX commands for including and rotating images commented out near the end of the LaTeX template.
- You should submit your work through the **class Gradescope 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). Failure to do so will result in your assignment not being graded.

Honor Code (Make Sure to Virtually Sign)

Problem HC.

- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

Agreed (I agree to the above, Blake Raphael). □

18 Standard 18: (Divide & Conquer) Basics and CounterExamples (4 points)

Problem 18. Given an array A of length n, we say it contains a duplicate if there are two distinct indices i and j (with $i \neq j$) such that $A[i] = A[j]$. Consider the following divide and conquer algorithm for testing whether an array has a duplicate.

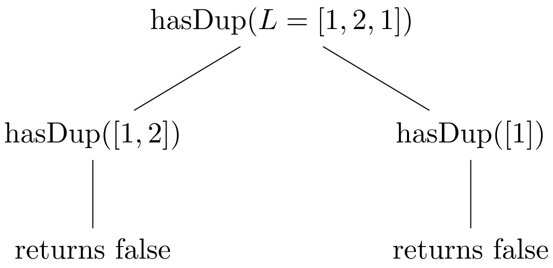
Algorithm 1 Check Duplicates Algorithm

```
procedure hasDup(List $A$)
    $n \leftarrow len(A)$;
    if $n == 2$ and $A[0] == A[1]$ then
        return true;
    else if $n > 2$ then
        return hasDup($A[0 : \lfloor n/2 \rfloor]$) or hasDup($A[\lfloor n/2 \rfloor : n]$);
    else
        return false;
```

Will the above algorithm correctly return true if A contains a duplicate and false if A does not contain a duplicate? **Explain and justify** your answer. If the algorithm does not correctly decide the presence of duplicates, give an example on which it fails, explain what the algorithm does on that example, and what the correct answer for that example is.

Answer. Consider the following list: $L = [1, 2, 1]$
This input will cause the algorithm to fail because the algorithm will return false and indicate there are no duplicates when there really are.

The following tree shows why:



The list gets split in half, and assuming we round down when dividing by an odd number for indices or else we would be stuck in infinite recursion, we see that one list has $[1, 2]$ and the other list has $[1]$. When hasDup is run again, the first list has length 2, but no duplicates, so we return false. The second time hasDup is run, we have a list of length 1, so we return false again. We then have an 'or' that chooses between the two falses to return, and returns false, missing the duplicate 1s in the original list.

□