

**CSCI 2300 — Introduction to Algorithms**  
**Lab 1 (document version 1.0) — January 15, 2020**  
**A Brief Look at Binary Search and Sorting**

- This lab has two checkpoints that are to be completed by the end of your assigned lab section. Show your code and results to a TA or mentor to receive credit. **Do not leave your lab section until you are checked off.**
- This lab is to be completed **individually**. Do not share your work or code with anyone else.
- You can use any programming language that you like; we suggest Python, C++, or C.
- For all of our labs, please avoid using Google to find suggestions or solutions. The goal is to use your own brain to work these problems out, which will help you develop the skills to do well on the exams and, more importantly, become a substantially better computer scientist.

1. Write pseudo-code for binary search, then implement it. More specifically, find whether a number exists in a given **sorted** list of numbers. And don't just call an existing library function to accomplish this. Test with different input datasets.

Are you sure you have no bugs in your code? Think about what pitfalls exist when translating an algorithm into code.

2. Given an **unsorted** and **immutable** array of size  $n - 1$  that contains all integers from 1 to  $n$  except one, identify which integer is missing. Assume you have a fixed amount of free memory that is much too small to hold  $n$  integers. Design and implement an algorithm to do this in  $O(n)$  time, which essentially means you can only “look” at each element at most a small constant number of times, ideally only once.

How many different  $O(n)$  algorithms can you come up with?