

# Assignment 3:

## Big-O, Searching, Hash Tables, and Sorting

### Overview

In this assignment, you'll apply what you've learned about Big-O notation, searching algorithms, sorting algorithms, and hash tables to answer a series of short answer questions.

### Submission

Assignment 3 is due on **November 5, 2021 at 11:59 pm PT**.

*Please note:* Assignment 3 is in the format of an iLearn quiz, and there will be no late submissions accepted. The quiz will allow unlimited attempts, and your last attempt will be accepted.

### Grading

100 points total:

- Question 1: 10 points
- Question 2: 10 points
- Question 3: 10 points
- Question 4: 10 points
- Question 5: 10 points
- Question 6: 10 points
- Question 7: 10 points
- Question 8: 15 points
- Question 9: 15 points

# Big-O Questions

All Big-O answers must be expressed in terms of simplified Big-O. To receive full credit on each question, you'll need to provide the correct answer **and** a detailed explanation of your reasoning in your own words. For example, you should explain not only the lines of code that result in your Big-O answer, but also the Big-O terms you were able to ignore when simplifying the expression. You can use the examples of Big-O analysis we went through in class as a reference for your how explanations should look .

## Question 1

What is the Big-O time complexity for the following method in terms of n?

```
public int process(int n) {  
    int result = 0;  
    for (int i = 0; i < n; i++) {  
        result += n;  
    }  
    for (int i = 0; i < n / 2; i++) {  
        result -= n;  
    }  
    return n;  
}
```

## Question 2

What is the Big-O time complexity for the following method in terms of N, if N is the size of words?

```
public int tally(ArrayList<String> words) {  
    int maxLength = 0;  
    for (int i = 0; i < words.size(); i++) {  
        String word = words.get(i);  
        if (word.length() > maxLength()) {  
            maxLength = word.length();  
        }  
    }  
    return maxLength;  
}
```

### Question 3

What is the Big-O time complexity for the following method in terms of N, if N is the size of numbers?

```
public int maths(ArrayList<Integer> numbers) {  
    int index = numbers.size();  
    int total = 0;  
    while (index >= 1) {  
        total += numbers.get(index - 1);  
        index /= 3; // same as index = index / 3  
    }  
    return total;  
}
```

### Question 4

What is the Big-O time complexity for the following method in terms of N, if N is the size of names?

```
public ArrayList<String> getPairs(ArrayList<String> names) {  
    ArrayList<String> pairs = new ArrayList<>();  
    for (int a = 0; a < names.size(); a++) {  
        for (int b = 0; b < names.size(); b++) {  
            String personA = names.get(a);  
            String personB = names.get(b);  
            if (personA.equals(personB)) {  
                continue;  
            }  
            pairs.add(personA + " & " + personB);  
        }  
    }  
    return pairs;  
}
```

# Search Algorithm Question

## Question 5: Search Algorithms

Explain, in your own words, what the tradeoffs are between using linear search and binary search. What are the pros and cons of each?

## Hash Table Questions

Suppose you have a hash table with 5 buckets, indexed 0, 1, 2, 3, and 4. We insert the following Strings, which have these hash codes:

- "Dasher" (hash code: 23)
- "Prancer" (hash code: 967)
- "Vixen" (hash code: 551)
- "Comet" (hash code: 9)
- "Cupid" (hash code: 1001)
- "Donner" (hash code: 77)
- "Blitzen" (hash code: 418)
- "Rudolph" (hash code: 90)

Assume we use the same techniques we described in lecture 12 for adding elements to a hash table. This hash table will be referenced in questions 6 and 7.

## Question 6: Hash Table Buckets

What Strings do each of the five buckets contain? Please explain how you determined your answers.

## Question 7: Hash Table Reshuffling

Suppose we increase the number of buckets in the hash table to 10, shuffling the same Strings into the buckets indexed 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Which Strings will end up in the same bucket they were in before? Please explain your reasoning.

# Sorting Algorithm Questions

## Question 8: Insertion Sort

Describe what kind of input would cause insertion sort to experience its worst case scenario. Please explain why it causes the worst case scenario.

## Question 9: Selection Sort

Suppose you apply selection sort to the following list of numbers:

9, 3, 11, 8, 14, 2, 7

After **three steps** of the algorithm, where each step involves passing through the list a single time, what would the list look like? Please explain your reasoning.