\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC249 Data Structure and Algorithms**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 03 **SEARCHING AND SORTING**

# Objectives

- Design and implement algorithms

- Understand and write code to do searching

- Understand and write code to do sorting

# Instruction and Problems

Write a Python program for each of the problems in this lab. Please use PyCharm to type and test your programs. Submit the Python files to Blackboard for credit.

## Program 1

Write a program to find the second largest item in a list. For example, in the list [2, 8, 7, 5, 4, 1, 6], the second largest item is 7. Sometimes the largest and second largest items are equal. For example, in the list [2, 8, 8, 5, 4, 1, 6], the second largest item is 8, which is equal to the largest item. The program is partially written. Your task is to define the function find\_second\_largest. Python has a max function, which returns the largest item in an iterable. You are **not allowed** to use the max function in this program. Also, Python has a sorted function and a sort method in lists, both of them sort items in an iterable. You are **not allowed** to use them in this program.

#

# Your name

# Date

# Finding second largest item in a list

#

def main():

list1 = [2, 8, 7, 5, 4, 1, 6]

second\_largest = find\_second\_largest(list1)

print("list1:", list1)

print("Second largest item in list1:", second\_largest)

print()

list2 = [2, 8, 8, 5, 4, 1, 6]

second\_largest = find\_second\_largest(list2)

print("list1:", list2)

print("Second largest item in list2:", second\_largest)

def find\_second\_largest(my\_list):

# add your code here

if \_\_name\_\_ == "\_\_main\_\_":

main()

Expected output:

list1: [2, 8, 7, 5, 4, 1, 6]

Second largest item in list1: 7

list1: [2, 8, 8, 5, 4, 1, 6]

Second largest item in list2: 8

Save your program in a file named **Lab03P1.py**. Submit it to Blackboard for credit.

## Program 2

The textbook shows you a few sorting algorithms. They are all comparison-based sorting algorithms. In this type of algorithms, items in a sequence are sorted by comparing their values. In this program, you are going to write a program to do a counting sort, which is a counting-based algorithm.

Counting sort is an easy and interesting sorting algorithm. It is efficient when there are multiple copies of the same values within a small range. It works by counting the frequency of each unique item within the range. It then uses the frequency counts to generate the sorted sequence. For example, suppose we have this list: [8, 2, 4, 5, 4, 7, 2, 6, 8, 6, 2, 5, 6]. There are three 2’s, two 4’s, two 5’s, three 6’s, one 7 and two 8’s. Within these frequency counts, we can generate this sorted list: [2, 2, 2, 4, 4, 5, 5, 6, 6, 6, 7, 8, 8].

This program is partially written. Your task is to define the function counting\_sort. This function has one parameter my\_list. It returns the sorted version of my\_list. Here is the algorithm:

1. Find the smallest item of my\_list.
2. Find the largest item of my\_list.
3. Create an empty list. This will become the sorted version of my\_list.
4. For each value v between smallest and largest, count the frequency of v. Use the frequency count of v to determine how many copies of v to be inserted into the sorted list.
5. Return the sorted list.

In your code, you can use Python’s max and min functions to find the largest and smallest items in my\_list. You can also use the built-in count method of my\_list to count the frequency of an item in the list. You are **not allowed** to use the sorted function or the sort method of lists in this program.

#

# Your name

# Date

# Counting Sort

#

def main():

list1 = [8, 2, 4, 5, 4, 7, 2, 6, 8, 6, 2, 5, 6]

list1\_sorted = counting\_sort(list1)

print("list1 before sorting:", list1)

print("list1 after sorting:", list1\_sorted)

def counting\_sort(my\_list):

# add your code here

if \_\_name\_\_ == "\_\_main\_\_":

main()

Expected output:

list1 before sorting: [8, 2, 4, 5, 4, 7, 2, 6, 8, 6, 2, 5, 6]

list1 after sorting: [2, 2, 2, 4, 4, 5, 5, 6, 6, 6, 7, 8, 8]

Save your program in a file named **Lab03P2.py**. Submit it to Blackboard for credit.

# Grading rubric

Program 1:

find\_second\_largest function [50 points]

Program 2:

counting\_sort function [50 points]