Metropolitan State University

ICS 140 Computational Thinking with Programming

Class Exercise 11

**Lecture Section**

1. What is the name of the search algorithm that loops over every item in the list to check if it contains the sought item?

Linear search

1. The binary search method is faster than searching all items but has an additional constraint. What is required of a list to use binary search?

Must be sorted

1. What does big O notation represent?

Upper bound runtime

1. Describe the Selection Sort algorithm.

Sorting one item at a time, finding the smallest element and moving to the front and so on

1. Describe the Insertion Sort algorithm.

Creates a new sorted list removing the old one as the new one is created

1. Describe the Bubble Sort algorithm.

Compares pairs of elements and swaps them accordingly

1. Describe the Merge Sort algorithm.

Cuts the list into sublist until each sublist has only one element than merges them as a sorted list

1. For the sort algorithms above, which one will generally be the most efficient as lists grown larger?

Merge sort

1. In addition to the increase in processing required as list grow, name another resource constraint to watch for.

Memory usage or consumption

1. Coding sorting algorithms requires a way to swap values between 2 places in a list. Choose one of the methods that can be used and write the python code for swapping values below.

**Coding Search Algorithms**

Write python function that implements the sequential and binary search algorithms. Use the provided search\_algorithms.py file provided to complete this exercise. In the search\_algorithms.py file, you will find some functions already built that will help test out the performance of the algorithms. I have also copied the pseudocode from the assigned reading to help simplify the coding process. Write functions with the names and inputs of the routines in the commented pseudocode. I would recommend copying the comments for each algorithm and then go through it line by line converting the pseudocode to python.

Note: Some lines of the pseudocode are not relevant to python. For example, there is no end if statement like there is in other programming languages. Lines like this can be deleted or commented out.

The output should like something like this:

Text

Description automatically generated

Smaller lists will be very close in speed. If we test out a larger number, we should see the binary search pull away with dramatically.

Text

Description automatically generated

Once you have coded the functions, paste the code for the 2 functions in their designated area below and take a few screenshots of your program running with different sizes of lists similar to the screenshots above.

**Selection Search Function code here**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Binary Search Function code here**

**A screenshot of a computer program

Description automatically generated**

**Screenshot of Test here**

**A screenshot of a computer screen

Description automatically generated**