# Data Structures and Algorithms

# INFO 6205

# Homework 6

# Due: October 24, 2020

Put all your java, compiled class files and documentation files into a zip file named Homework6.zip and submit it via the dropbox on canvas before the END of due date.

Put your name on all .java files. There will be a short quiz on this homework.

1. Consider a[] = {91, 37, 42, 38, 3, 9, 62, 10, 21, 8, 34, 19, 6, 18, 21, 25}

Walk through QuickSort algorithm: a) pivot Last-element; b) pivot First-element;

c) pivot Randomized

A) Show step by step how the sorting works

B) Write Java code, compile and run

C) Time and space complexity

D) Compare the results for pivots a, b, c

1. Consider a[] = {91, 37, 42, 38, 3, 9, 62, 10, 21, 8, 34, 19, 6, 18, 21, 25}

Walk through InsertionSort algorithm:

A) Show step by step how the sorting works

B) Write Java code, compile and run

C) Time and space complexity,

D) Compare the results of InsertionSort with results of QuickSort in p1

1. Consider a[] = {91, 37, 42, 38, 3, 9, 62, 10, 21, 8, 34, 19, 6, 18, 21, 25}

Walk through TimSort algorithm:

A) Show step by step how the sorting works with MinRun=32

B) Write Java code, compile and run

C) Time and space complexity

D) Compare results of TimSort with results of InsertionSort in p2

E) Compare results of TimSort with results of QuickSort in p1

1. Consider a[] = {91, 37, 42, 38, 3, 9, 62, 10, 21, 8, 34, 19, 6, 18, 21, 25}

Walk through BubbleSort algorithm:

A) Show step by step how the sort algorithm works

B) Write Java code, compile and run

C) Time and space complexity

5. Consider attached image Boston.jpg. Write a program to sort the image Pixels by “brightness”. using three sorting algorithms: QuickSort, InsertionSort, TimSort You need to sort the Pixel array size of the image and show the runtime complexity of each Sorting algorithm and compare. You may NOT use any Java library function for sorting. You should use only the Sorting Java code I provided in class. The Pixel sorting should start from (0,0) to (high, high) for Brightness. For each Pixel, you need to convert RGB color to appropriate intensity. Use intensity formula:

I = 0.2989R + 0.5870G + 0.1140B; If the current pixel Intensity is larger than the next pixel intensity, you need to swap, going in Ascending order.

6. Sort string step-by-step: a) QuickSort, b) InsertionSort, c) TimSort;

Write Java code for each algorithm

7. Table shows the performance of different Sorting algorithms. Consider only the worst-case running-time and worse-case space complexity. A) Explain the running-times complexity for all listed algorithms and differences between them; B) Explain the space complexity for all listed algorithms and differences between them;