# Data Structures and Algorithms

# INFO 6205

# Homework 5

# Due: October 17, 2020

Put all your java, compiled class files and documentation files into a zip file named Homework5.zip and submit it via the drop box 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 this MergeSort recursive algorithm with termination point r>1 with two MergeSort and Merge methods.

A) Provide a diagram that shows the entire Stack operations. For example: you start

with step1 and then step2 to handle the first-half and then second-half, at what point

step3 starts? At what point step4 starts? Note: This is recursive calls

MergeSort(arr[], l, r)

If r > l

1.Find the middle point to divide the array into

two halves: middle m = (l+r)/2

2.Call mergeSort for first half:

Call mergeSort(arr, l, m)

3. Call mergeSort for second half:

Call mergeSort(arr, m+1, r)

4.Merge the two halves sorted in step 2 and 3:

Call merge(arr, l, m,

B) Consider the following input Data:

{91, 37, 42, 38, 3, 9, 62, 10, 21, 8, 34, 19, 6, 18, 21, 25}

a) Sort data Graphically, show step-by-step, recursion on Stack

and what is Termination point?

b) Write Java code and Compile and Run with provided data.

2. Consider HeapSort algorithm with input array data =

{91, 37, 42, 38, 3, 9, 62, 10, 21, 8, 34, 19, 6, 18, 21, 25}

A) Show step-by-step algorithm to sort data in Descending order:

a) MinHeapify, b) Replace one element

B) Consider example code: <https://www.studytonight.com/data-structures/heap-sort>

Is this code sort in Ascending order or Descending order? Explain

C) Write Java code for the algorithm with input data to sort in Descending order

D) Is the output result is the same as your program in problem-1? Explain

E) Show step-by-step Algorithm to sort in Ascending order that is steps:

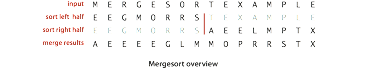
a) MaxHeapify, b) Replace one value

F) Change the program to sort in Ascending order

G) Is the output result is same as output for your program in problem-1? Explain

3. Consider attached image Boston.jpg. Write a program to sort the image Pixels by “brightness”. Write program for four sorting algorithms: HeapSort and MergeSort. You need to sort the Pixel array size of the image in Ascending order 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.



4. Consider mergeSort algorithm for the following input string. Show the stack operations push and pop step by step for call mergeSort(arr, l, m) and call mergeSort(arr, m+1, r). Note: I don’t need the entire program, just show step by step stack push and pop operations, the recursive Tree structure

5. The following table is the Truth table for bitwise AND, OR, XOR operations.

One application of XOR is to perform as a parity-checker on the network where sending and receiving data packets, and want to make sure the data is correctly received.

a) Write the range of X binary number starting from zero

b) Compare the range of binary numbers, identify the number of bits differences.

c) Binary value for X=16=10000

d) Convert X binary to gray binary

E) Convert gray binary to regular binary

c) Suppose X binary = 0110010111011101

d) Convert Binary X to Gary binary

e) Convert Gray binary to regular Binary

6. The following dynamic programming algorithm generates Gray code. The results of this gray code shown in Figure1, the jth column of gray code binary string of length n is actually stored in the vertical block starting with 2 j+1th index of (j+1)th column of (n+1)-lengthpasted-image.tiff ordinary binary code. Exploiting this pattern, the following loop is used to generate gray code with a dynamic programming sense.

*for i 1 to 2n do*

*for j 1 to n do*

*if (i mod 2j+1 = 2j ) g[j] (g[j] XOR 1);*

Walk through Figure1 to describe dynamic programming relationship between decimal,

binary, and binary gray code:

Figure1: d=decimal, b=binary, g=gray