# Data Structures and Algorithms

# INFO 6205

# Homework 5

# Due: Wed March 23, 2022

Put all your java, compiled class files and documentation files into a zip file Homework5.zip

and submit it via Canvas before the END of due date. Put your name on all .java files.

1. Identify heap algorithms, write the code for each algorithm and explain step by step how

each algorithm works: NOTE: you need to read book section Priority Queues

A) If the heap order is violated because a node's key becomes larger than that node's parents key, then we can make progress toward fixing the violation by exchanging the node with its parent. After the exchange, the node is larger than both its children (one is the old parent, and the other is smaller than the old parent because it was a child of that node) but the node may still be larger than its parent. We can fix that violation in the same way, and so forth, moving up the heap until we reach a node with a larger key, or the root.

B) If the heap order is violated because a node's key becomes smaller than one or both of that node's children's keys, then we can make progress toward fixing the violation by exchanging the node with the larger of its two children. This switch may cause a violation at the child; we fix that violation in the same way, and so forth, moving down the heap until we reach a node with both children smaller, or the bottom.

C) Suppose that your application will have a huge number of insert operations, but only a few remove the maximum operations. Which priority-queue implementation do you think would be most effective: heap, unordered array, ordered array?

D) Suppose that your application will have a huge number of find the maximum operations, but a relatively small number of insert and remove the maximum operations. Which priority queue implementation do you think would be most effective: heap, unordered array, ordered array?

E) What is the minimum number of items that must be exchanged during a remove the maximum operation in a heap of size n with no duplicate keys? Give a heap of size 15 for which the minimum is achieved.

2. Consider the following, Input Data: {54, 18, 77, 24, 11, 27, 43, 38, 3, 9, 82, 10, 21, 8, 34, 19, 6}

a) Graphically build a Circular queue for input data. Discuss and show Head

and Tail pointers at each step:

i) enqueue all input data

ii) dequeue 8 elements

iii) enqueue 5 elements

iv) dequeue all elements

b) Write Java code for the Circular queue, provide enqueue, dequeue, isEmpty, isFull,

and displayQueue methods, to show the status of the queue with steps in (a).

Compile code and Run with input data.

3. Provide computations:

a) Convert Binary to Hex, 1011010010110101

b) Convert Decimal 128 to Binary

c) Convert Hex 0x4B7D to Binary

d) What is the ASCII character for these numbers?

0x3B, 0x00, 0x69, 0x31, 0x20, 0x64, 0x4D, 0x20, 0x66, 0x33, 0x3C, 0x21

e) Describe signed and unsigned numbers

4. Consider String: “Students want to transfer files with gmail”

note1: The algorithm uses frequency of text data elements to generate data compression;

note2: You don’t need to use the actual ASCII values; you can initialize characters: 1,2,3..

a) Generate a binary Huffman Tree step-by-step

b) Show binary data both before and after compression. Analyze difference.

c) Implement the following Huffman Encoding Algorithm

Compile Java code and run it with the input string provided above

d) Show as to why/how the algorithm uses PriorityQueue

Begin

   define a node with character, frequency, left and right child of the node for

Huffman tree.

   create a list ‘freq’ to store frequency of each character, initially, all are 0

   for each character c in the string do

      increase the frequency for character ch in freq list.

   done

   for all type of character ch do

      if the frequency of ch is non zero then

         add ch and its frequency as a node of priority queue Q.

   done

   while Q is not empty do

      remove item from Q and assign it to left child of node

      remove item from Q and assign to the right child of node

      traverse the node to find the assigned code

   done

End

5. Consider the following code for User class.

A) What does the code do?

B) Explain methods: equals, hashcode, compareTo

C) How Comparable<User> is used?

C) Write Java code to test User class with multiple test data cases to test

equals, hashCode and CompareTo methods.

public class User implements Comparable<User> {

private String name;

private int id;

private Date birth;

public User (String name, int id, Date birth)

{ this.name = name; this.id = id; this.birth = birth; }

@Override

public boolean equals(Object other) {

if (this == other) return true;

if (other == null || (this.getClass() != other.getClass()))

{ return false; }

User guest = (User) other;

return(this.id == guest.id) &&

(this.name = null && name.equals(guest.name)) &&

(this.dob != null && dob.equals(guest.birth));

}

@Override  public int hashCode() {

int result = 0;

result = 31\*result + id;

result = 31\*result + (name !=null ? name.hashCode() : 0);

result = 31\*result + (birth !=null ? dob.hashCode() : 0);

return result;

}

@Override

public int compareTo(User o) {

return this.id - o.id; }

}

}

6. Java String hashcode is the following: pasted-image.tiff

What is the hashCode 32-bit integer number for string =“Welcome Students”?

A) Mathematically by hand,

B) Write Java code