Sean Morrissey

Professor Ramoza Ahsan

CS 2223 Algorithms

4 April 2019

Assignment 3

Question 1.

```
Function MergeKElements(List1,…ListN)
    Create Result Array of size N*K elements
    Build a min heap of size k
    Insert first element of each array into heap
    Loop until there are no elements left to sort while doing the following:
        Remove the root of the heap and amend it to the result array
        Insert the next smallest element from the same list that the previous min element
        was deleted from
    Return the result array
```

Creating the heap is O(k) time, deleting the min element from the heap is O(log k) time, and inserting the smallest element from that same list is also O(log k) time; therefore, the total time is O(k + n log k) time which is the same as O(n log k) time.

Question 2.

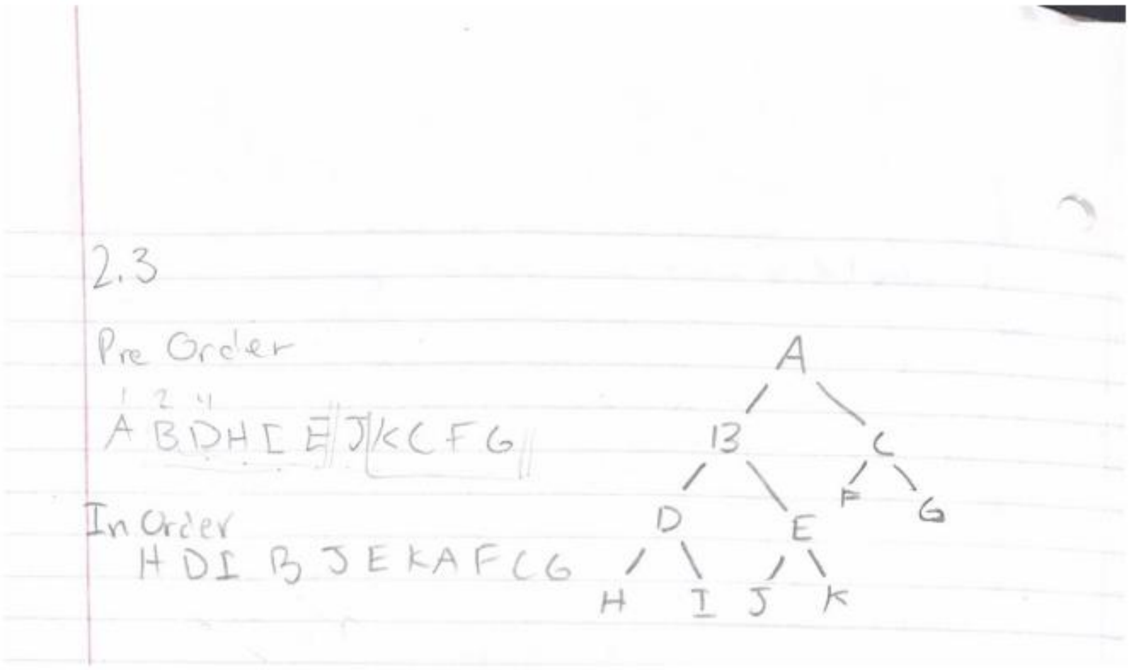
2.1 Group C and group E would not be examined. In group C, 912 is greater than 911 which does not follow the rules of binary search trees. Similarly, in group E, 278, which is placed early in the sequence, is less than 299, which is later in the sequence. This breaks the rules of binary search trees as well.

2.2

```
Function validBST (Binary Search Tree)
    Get root node
    Create a min constraint integer
Create a max constraint integer
Function isBST(node, min, max)
    If the node is empty
        Return true
    If the node's data is less than min or greater than max
        Return false
    Else
        Return the Boolean value of ( isBST recursively on the left and
        right hand sides of the trees)
```

The recurrence function of validBST() is T(N) = T(N) + c, so the big o runtime , according to the master theorem, is O(N).bv

2.3



2.3

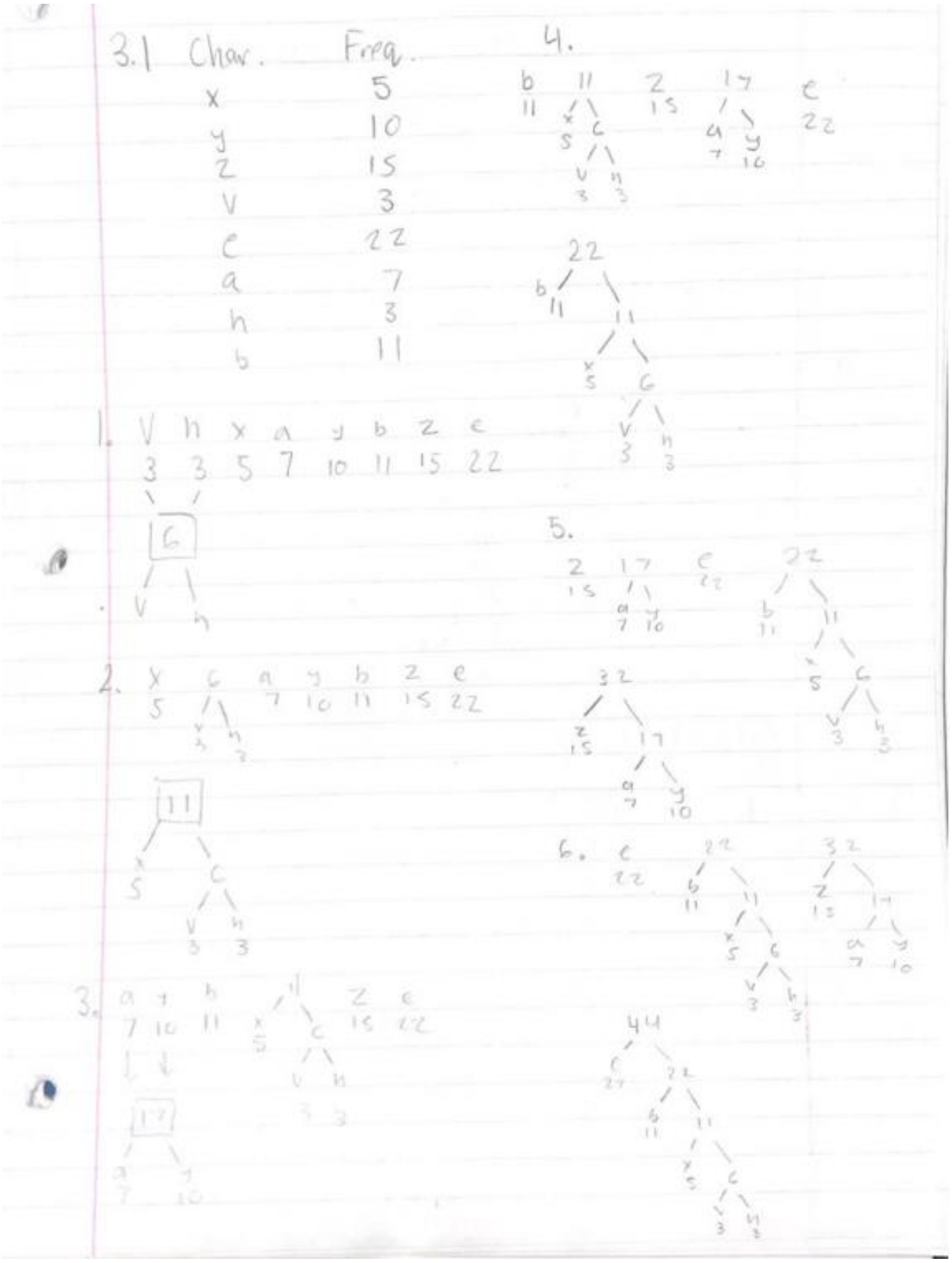
Pre Order

1 2 4
A B D H I E J K C F G

In Order

H D I B J E K A F C G

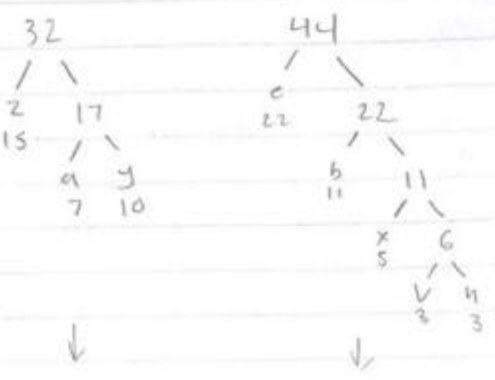
Question 3.



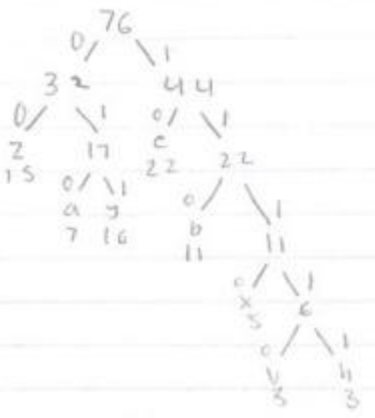
3.1

Char.	Freq.
x	5
y	10
z	15
v	3
e	22
a	7
h	3
b	11

7.



Huffman Tree:



Coding Table

Char.	Code
z	00
a	010
y	011
e	10
b	110
x	1110
v	11110
h	11111

3.2 "habbhxzyehv"

11111 010 110 110 11111 1110 00 011 10 11111 11110