Data Structure

Homework 3

Deadline: 2021/12/14 Tue. 23:55

Task 1: (I/O: 35 points, coding style: 5 points)

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code.

1. Build a Huffman Tree from input characters.
2. Traverse the Huffman Tree and assign codes to characters.

While building the Huffman tree by bottom-up method, please follow the rules below:

(1) If more two different characters have the same frequencies, order them by their ASCII code.

(2) The value of the right node has to be larger than or equal to the one of the left node. If the character have the same frequencies with the combination, put combination to the right node.

(3) Follow the order in (1) to read the value, and build the tree from the left subtree.

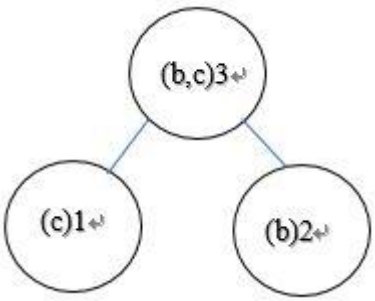
(4) The values of left and right branches are “0” and “1”, respectively.

Example:

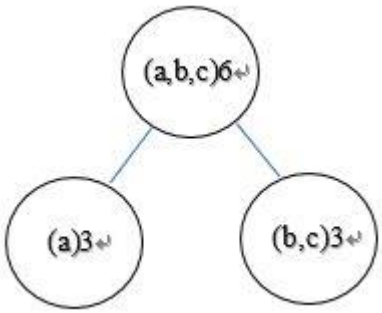
Input: `abcaba`

Character:	a	b	c
Frequency:	3	2	1

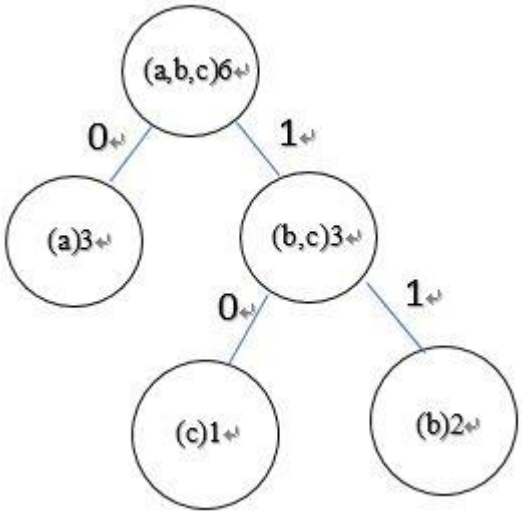
Step1: Extract two minimum frequency nodes. Add a new internal node with frequency.



Step2: Extract two minimum frequency nodes just like step1.



Step3: Use Huffman coding to express what you assign codes to characters.



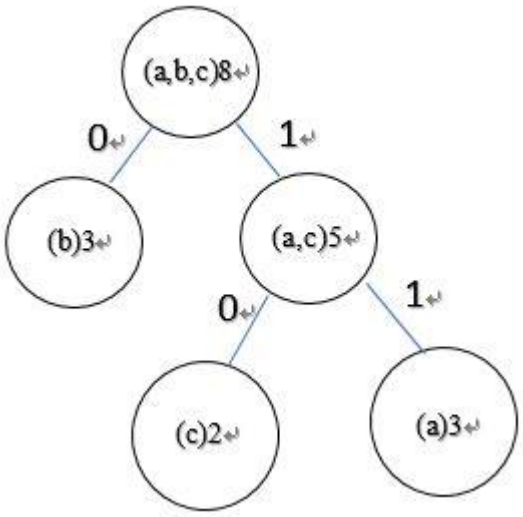

```
Huffman Codes are :

b 11
a 0
c 10
```

Step4: Given that the root is level 1, the minimum weighted external path length is:
1*3+2*2+2*1 = 9

If the character with the same freqenecy, then choose the node with lower ASCII value.

For example:

Input: `aaabbbcc`



The minimum weighted external path length is: 2*3+1*3+2*2 = 13

The program must be implemented **by linked list**, or you will get zero points.

Example:

Input	Output
abcaba	> 011100110 > 9
aaabbbcc	> 1111110001010 > 13

Task 2: (I/O: 35 points, coding style: 5 points)

Given a number n (n = -1 or $1 \le n \le 2^{10}$) represents n integers. The next row is given n serial integers ($-2^{31} \le$ values $\le 2^{31} - 1$), representing the nodes. Please write a program to build a binary search tree and show the preorder and postorder traversals. The root is the first integer of the serial integers. If the upcoming integer equals to an existing node, put it on its right (right-child). For other cases, put the smaller ones on left (left-child), bigger ones on right (right-child). The program must be implemented **by linked list**, or you will get zero points.

Example:

Input	Output
8 7 3 4 8 2 6 5 1	> Preorder: 7 3 2 1 4 6 5 8 > Postorder: 1 2 5 6 4 3 8 7
3 100 1 10	> Preorder: 100 1 10 > Postorder: 10 1 100

Put the files below in the folder (folder name: studentID), and compress this folder as **"studentID.zip"**.

1. Two source code files (filename: studentID_1.c, studentID_2.c)
2. One report with your coding environment (OS, IDE, ...), problems you encountered, and references. (filename: studentID.pdf) (10 points)

All the file names are correct, or you'll get zero points. (10 points)

You must hand in the assignment on time, or you will get zero points.

Warning: We encourage you to discuss assignments with each other. However, you have the responsibility to finish the assignments individually. **Do not copy others' assignment, or you will get zero points.**