timallec

Name, SID, Date

In Class Assignment 25: Huffman Encoding

Benjamin Sanders, MS November 25, 2020

1 Introduction

You may work in groups of up to two or three students. Write all student names at the top of all pages for this assignment. Turn in all work to Blackboard on or before the deadline to receive credit.

You may use additional libraries and online resources, if you get them approved in writing, over email, from the instructor first. If you have received approval from the instructor, write the approved libraries and any references in the space below.

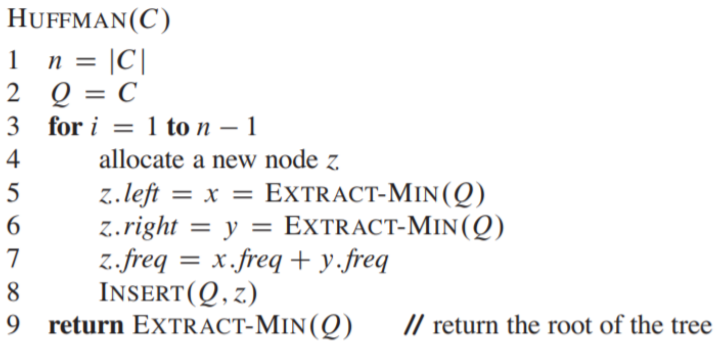
2 Assignment Description

2.1 Big Picture

This is the lossless algorithm that is used in compression methods, such as those used to produce *.zip files.

2.2 Algorithm Implementation

Implement the following algorithm in Java, using the Vector data structure for any 1-D array, 2-D array, or linear algebra purposes.



```
HUFFMAN($C$)
$n = |C|$
$Q = C$
for $i = 1$ to $n - 1$
    allocate a new node $z$
    $z.left = x = $ EXTRACT-MIN($Q$)
    $z.right = y = $ EXTRACT-MIN($Q$)
    $z.freq = x.freq + y.freq$
    INSERT($Q, z$)
return EXTRACT-MIN($Q$)    // return the root of the tree
```

Note that C is a set of n characters. Note that Q is a min-priority queue. If you did not get your min-priority queue working yet, you may use an external code library for this purpose.

2.3 Time Complexity Analysis

Where n is the number of characters in C, analyze the time complexity of the given algorithm with respect to n. Write the result of your analysis in big-O notation, i.e. $O(n^2)$ in the space below.

O(n^2)

2.4 Space Complexity Analysis

Where n is the number of characters in C, analyze the space complexity of the given algorithm with respect to n. Write the result of your analysis in big-O notation, i.e. $O(n \cdot log(n))$ in the space below.

O(n)

Name, SID, Date ....

3 New Algorithm Design and Implementation

In the space below, design a new algorithm to perform lossy compression, like what is used to create *.jpg files. Use pseudocode written in a style similar to the given algorithm, and implement it in Java. You may use as many additional pages as necessary for this purpose.

4 What to Turn In

Turn in one PDF or Word document on Blackboard, containing the following items.

1. All pages scanned or photographed of the In Class Assignment completed document.
2. Any additional pages you used to complete the assignment.
3. All code created for the assignment, along with test cases.
4. One statement indicating which parts of your implementation(s) are working, and which parts are not.
5. Screenshots demonstrating the code working, if it is working.