Software Development Report
for
CSC 122: Programming II: Fall 2022
Huffman
by
Manel Casado

Problem Summary

The program consists in doing lossless compression. The program has to decode an encrypted file using a Huffman Coding

Implementation Requirements

- Binary trees
- Huffman coding
- Recursion
- Decoding
- Read files

System Design

Testing Plan

Case	Description	Input	Expected output
S1	Scan a file with a small amount of data	File through scanner	Read and scan the data
S2	Scan a file with a medium amount of data	File through scanner	Read and scan the data
S3	Encode file testing.dat	encode	Encoding: abcdabafac

		testing.dat (a b c d a b a f a c)	Leaf nodes before sort: a: repeats: 4 path: b: repeats: 2 path: c: repeats: 2 path: d: repeats: 1 path: f: repeats: 1 path: Leaf nodes after sort (non mandatory but improves performance: d: repeats: 1 path: f: repeats: 1 path: b: repeats: 2 path: c: repeats: 2 path: a: repeats: 4 path: Huffman tree: : repeats: 10 path: a: repeats: 4 path: 0 : repeats: 6 path: 1 c: repeats: 2 path: 10 : repeats: 4 path: 11 : repeats: 2 path: 110 d: repeats: 1 path: 1100 f: repeats: 1 path: 1101 b: repeats: 2 path: 111 Characters with its path: a : 0 c : 10 d : 1100 f : 1101 b : 111 Characters coded in file testing.encode: 01111011000111011010 10
S4	Decoding testing.encode which contains the testing.dat coded before	decode testing.decode	: repeats: 0 path: a: repeats: 0 path: 0 : repeats: 0 path: 1 c: repeats: 0 path: 10 : repeats: 0 path: 11 : repeats: 0 path: 110

			d: repeats: 0 path: 1100 f: repeats: 0 path: 1101 b: repeats: 0 path: 111 Decoded: abcdabafac
S5	Encoding file testing2.dat	Encode Testing.dat (a b r a c a d a b r a d)	Encoding: abracadabrad Leaf nodes before sort: a: repeats: 5 path: b: repeats: 2 path: r: repeats: 2 path: c: repeats: 1 path: d: repeats: 2 path: Leaf nodes after sort (non mandatory but improves performance: c: repeats: 1 path: b: repeats: 2 path: r: repeats: 2 path: d: repeats: 2 path: a: repeats: 5 path: Huffman tree: : repeats: 12 path: a: repeats: 5 path: 0 : repeats: 7 path: 1 : repeats: 3 path: 10 c: repeats: 1 path: 100 b: repeats: 2 path: 101 : repeats: 4 path: 11 r: repeats: 2 path: 110 d: repeats: 2 path: 111 Characters with its path: a : 0 c : 100 b : 101 r : 110 d : 111
			Characters coded in file testing.encode: 010111001000111010111100111

S6	Decoding testing.encode which contains the testing2.dat coded before	decode testing.decode	: repeats: 0 path: a: repeats: 0 path: 0 : repeats: 0 path: 1 : repeats: 0 path: 10 c: repeats: 0 path: 100 b: repeats: 0 path: 101 : repeats: 0 path: 11 r: repeats: 0 path: 110 d: repeats: 0 path: 111 Decoded: abracadabrad
Err1	Type something that is not “encode” or “decode” after the first question	a	Question asked again: Do you want to encode or decode? a Do you want to encode or decode?
Err2	Type a wrong or incomplete filename after asking for an encode	testing	File not found. Encoding: Leaf nodes before sort: Leaf nodes after sort (non mandatory but improves performance: Huffman tree: Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Node.getLeftNode()" because "node" is null
Err3	Type a wrong or incomplete filename after asking for an encode It always decodes testing.decode	a	: repeats: 0 path: a: repeats: 0 path: 0 : repeats: 0 path: 1 : repeats: 0 path: 10 c: repeats: 0 path: 100 b: repeats: 0 path: 101 : repeats: 0 path: 11 r: repeats: 0 path: 110 d: repeats: 0 path: 111

			Decoded: abracadabrad

Time Spent

The program and development report took me around 19 hours to develop

Outside resources used

- UML cheat sheet provided in Canvas
- Lab code and lecture examples from Google Drive
- Microsoft Visio to create the System Design
- Help from my dad designing, testing and fixing errors

Security Report

- Positive: If the decoding fails, no harm is done to the file
- Negative: Not being able to read the file

Ethical Report

- I can't think of any ethical risk since this program doesn't use any sensitive data
- A positive application could be decoding important files coded before to not get information stolen. This could be used by many businesses or associations.

Future improvements

- Handle the exception when not typing a proper file name after asking for an encode
- Scan the file by the given name when decoding and not scan only testing.decode

Lessons learned

- How Huffman Coding works
- Binary Trees

Improvements of Work

- Better testing