Huffman Encoding & Decoding

Midpoint Deliverable

By Jake Sellers & Patrick Motard

Github Repository Link

2. Summary:

We have an AngularJS web application that can be run by opening `index.html` in Firefox. If you launch `index.html` in a browser other than Firefox you may face CORS issues. The IDE we use is sublime text 3. It is lightweight and straightforward. Works on linux/windows/mac. The bulk of our code has been written in PLC_FinalProject/home/homeController.js. Within homeController.js you will find the following methods:

listGenerator : This function is a stand-in until we integrate PEG.js into the project. It takes a string and returns and array of words.

listCount : Takes in an array of words and returns an array of objects. Each object has a 'word' and 'freq' property. There are as many objects as there are distinct words in the input string. The frequency of each object is equal to the number of times 'word' for each object occurred in the text.

orderByFreq : Sorts and array of {word, freq} by freq and returns the sorted array

Node : predefined node object for tree data structure

PriorityQueue (not fully implemented) : Will have an insert and combine method.

runTests : task runner for all logged test

TestOrderByFreq: Tests OrderByFreq

encodeGrammar.pegjs : PegJs grammar file. Will parse any amount of text and return an array of words in string format. Although we haven't integrated peg into our project, you can test out the grammar on pegs online parser generator found here. Paste the contents of encodeGrammar.pegjs into box 1 in their online tool. Then start typing text in box 2. View Results in box 3.

3. Justification :

Things are going very well with the project. We're ahead of our midpoint goals submitted in our proposal. It's coming along well.

4. Compilation:

1. cd to PLC_FinalProject

2. open index.html in Firefox

3. Firefox -> f12 (opens developer console)

4. Developer console -> click "Debugger" tab

5. Select homeController.js

6. Set breakpoint by clicking on margin to left of line the code view

7. press f5 to reload the page, if the breakpoint is hit, the page will freeze and wait for your next command

8. Step commands are on top left of developer console

5. Test Cases

Patrick made a mini testing framework and implemented it to show an example to be utilized for future testing. TESTOrderByFreq is implemented. It is an instance of the Test object and will test whether or not OrderByFreq successfully sorts an array by the expected parameters.

All tests which are queued in the runTests function are run on page load in the init() function. All tests that run will push a TestResult to vm.testResults. You can view the test results under the 'Test Results' section. If you would like to confirm that TestOrderByFreq passes by stepping through the code yourself do the following:

1. Place a breakpoint on line 95 of homeController.js in the Debugger window of Firefox.
2. Add a watch expression for vm.testResults so you can see its value before and after the test is run. To do this, right click and select "add selection to watch expression". Type in vm.testResults and press enter. It will now show up in your watch variables along the right.
3. Press f10 to step over the call or press f11 to step into the function and watch its behavior step by step.
4. View the TestResult object passed into vm.testResutls.

6. Code Walk-Through:

Code is explained in the Summary section and the Test Cases section.

7. Issues:

No issues so far. The decode grammar is going to be interesting to implement. We would like some clarification. This is our perception of the parsers role in the project. Please correct us wherever we misunderstand.

Input: “hi hello hey hi” -> parser encode -> Output: [‘hi’, ‘hello’, ‘hey’, ‘hi’]

Input: “!0<-hi01<-hello11<-hey 0101110” -> parser decode -> Output: “hi hello hey hi”