Assignment Solution

This is a detailed note on how I approached this assignment . The primary language of the assignment was Scala.

There are 3 primary classes
- Entity
- Feed
- Convert

There are 2 objects
- Main
- Test

Implementation of the solution as follows:

Input: Feed:Obama visited Facebook headquarters: http://bit.ly/xyz @elversatile

Entry Class : The input feed is given and the (Tag,start_position,end_position) are given . I call this collection an entity and wrap it in an Entry class. This is the granular most structure and forms the basis of my algorithm.

Feed_in class: This is the Feed Data Structure. It contains the String and the array of entry classes which now defines how the input is stored.

It has the following definitions:
- Add_to_list: It can add any external Entry to a list
- Show Contents: It shows the contents of the list
- Find Overlaps: It finds if there is a flaw in the user input. Described in the later section

Convert class: This is the main structure supporting all primary algorithms. It converts the feed in an O(n) with a single parse of the entity list which itself takes a time of O(mlogm) to form and sort . Another aspect of this class that there is an encoder which encodes string patterns to an array. Let us describe each one in Detail:

Data Structures:
- Hashmap with <key,value> = <"Tag", Array of Strings>

Examples: "Entity" -> Array("N","<strong>","","","</strong>")
"Link" -> Array("Y","""<a href="""","",""""">""","</a>")

The reason I chose an array is that I noticed that there was a fixed pattern in every initialization of html tags. “<a href =” is followed by a custom input followed by closing of a href which is “>” . The presence of this input defines the first element of the array which is a “Y” or an “N”.

String Encoder: A normal person would face a huge problem if someone has to actually store the string in a way I described above. Hence , I designed an encoder for users simplicity so that a person can actually enter the pattern in a convention he/she wants. Let me give a few examples:

1. Px<a href="INPUT">Sx</a> **------>** (Y, <a href=", , ", </a>,)
2. Px<strong>Sx</strong> **------>** (N, <strong>, , , </strong>,)
3. Px<a href="http://twitter.com/INPUT">Sx</a> **------>** (Y, <a href="http://twitter.com/, , ", </a>,)

The convention is as follows:

- Px stands for Prefix, Sx stands for Suffix, I have used regular expressions for detecting these locations . Anything that has to be entered as a prefix has to be put after prefix and anything that has to be put in the suffix has to be entered after Sx.
- INPUT stands for custom input and that is how the algorithm knows when to add the input from the substring that is already provided in the Entry tag. As an example <a href="http://twitter.com/INPUT> becomes a <href="http://twitter.com/elversatile>. I use regular expression finding again to detect it in the encoder.

String Decoder: This just decodes an element into its encoding. The reason I created this module is that if someone is testing the application, The person can retrieve the encoding from the hashmap and retrieve the code backwards if needed.

Core Algorithm: The function convert1 uses a new stringbuilder and that is used to append the words and create the resulting output.

Out of Bounds: This is just a test we should run before starting the algorithm . It tests for all the wrong entries in the feed.

Tests

1. Test1 : Testing the functionality. I have just taken the example in the assignment and shown it works!!!
2. Test2: Encoding: Check encodes and decodes to find out the conversions are right or wrong.
3. Test3 Check overlaps: We can have a situation where we can have (Entity,1,10) and Link(5,15). This is an overlap situation and we do not handle it at this stage of the application.
4. Test4 : Index out of bounds: A very common mistake. Developers often use indices more than the Feed size. This test will prevent them from making mistakes
5. Test 5 : Adding a new Tag like Hashtag . I have taken a new hashtag example to show that if the Pattern match is followed, a hashtag can be introduced as given in the problem statement.