Lab #2 – paired tags analysis of a *html* file

CSC221 – Spring 2020

Due date: February 25, 2020, 5pm

Hyper Text Meta Language (*html*) provides a rich language for representation. *html* is used to represent text, images, sound, etc. Even though *html* is not a programming language, it is a language with *grammar* rules. *html* supports a number of grammatical tags[1] which must be paired with a corresponding *begin* and *end* tags[2]. For example, the pair of tags `<bf>` and `</bf>` can be used to indicate bold font text such as

```
<bf>This is shown in bold font.</bf>.
```

Another pair of tags is `<it>` (italics), and can be used as

```
<bf> This is shown in bold font and <it> this bold italics
font</it>.</bf>
```

The grammar rule for for matched *html* tags is similar to that of parenthesized expressions, they must match. Compliance with this rule can be easily checked using a stack based algorithm.

This is a pledged work assignment. The work you submit for grading **must** be the result of your efforts. You may consult only with me and the graduate student teaching assistant on the details of your assignment. You may discuss with each other syntax issues and algorithmic ideas that have been discussed in class. If you have any question about whether discussion with others is appropriate, you should consult with me.

This assignment requires you **(1)** to write a Java program that will read a set of paired *html* tag names from a file[3], **(2)** to store the *html* tags in a (tree based) Set, **(3)** to read an *html* file by interactively specifying an Uniform Resource Locator (URL), **(4)** to check for compliance of our "tag grammar" rule using a Stack, and **(5)** finally to report if the *html* file is in compliance or not. You **must** use the Jsoup library[4], the Java Stack class, the Java Set interface with the Java TreeSet class, and the Java RegEx class.

One of your tasks will be to search an *html* file for a tag. We will use a regular expression to accomplish this. The regular expression you should use is:

[1] A list of html tags can be found by searching for "html paired tags".

[2] *html* also supports tags that are not paired.

[3] At least 15 tag names.

[4] Provided as part of the gitHut Classroom distribution.

```
"<(/?)([a-zA-Z]+)"
```

This regular expression will match a string which

1. starts with the character `<`,
2. optionally is followed by the character `/`, and
3. finishes with a span of letters.

Notice that our regular expression does not include the final `>`, there is a good reason. Strings that are matched by our regular expression include: `<bf`, `</it`, and `</CS`. The parentheses in the regular expression correspond the matching groups, `group(0)` matches the entire string, `group(1)` would match either '/' or the empty string, and `group(2)` would match the tag name. The way in which we want to use the regular expression is:

```
import java.util.regex.*;

Pattern p = Pattern.compile(”<(/?)([a–zA–Z]+)”);
String target =
Matcher m = p.matcher(target);

while(m.find()) {
    System.out.println(m.group(0));
    System.out.println(m.group(1));
    System.out.println(m.group(2)};
```

These snippets will be explained much more in class.

The Stack based algorithm to verify the matched tags goes something like

```
as long as I can find a tag (m.find())
   token2 = m.group(2)
   token0 = m.group(0)

   if token2 is not in my set of tags
      look for another tag

   if token0 is an "open" tag
        push token0 on the stack
    else if token0 "matches" an open tag at the top of the stack
        pop the stack
    else
        push the "closed" tag token0 on the stack
```

Your Java program must be well written and readable. Readability includes appropriate indenting, spacing, variable naming, and commenting of code. Points will be assessed if your Java program does not comply. You must place your name in each *.java file in a comment at the top of the file.

This assignment is due no later than 5pm on Tuesday, February 25. You must submit your work for grading through gitHub classroom.