Introduction

Here’s a little task for you to show your skills. We respect your time and understand that applying for jobs is a full-time job. That’s why we tried to design a task that you can do in one evening. It shouldn’t take you longer than three to five hours. We don’t ask you to spend longer than that. Of course you’re free to invest as much time as you see fit. The goal, though, is not to “complete” the exercise, but rather to show us how you would approach a very specific challenge.

Task

Summary

Create a Java method that re-inserts HTML tags in a string at suitable positions - see details below.

Expectations

Git is the source code management tool used for your backend. You may use such online services such as Github, Gitlab and Bitbucket. When using Gitlab: please make repositories private and then share with us by adding the following GitHub users: robertgrabowski, cbrunzema, kijimuna (if you don’t have a GitHub account, please let us know, and we’ll find a way).

While the core task is only about defining a method implementation, please provide a **Readme** file that describes how we can compile the program (e.g. with Maven) and we can invoke the method with different parameters. For example, you may want to write a simple Java program that takes the two method parameters from the command-line, or alternatively a web service that provides a REST API endpoint which takes the two parameters - whatever you find easier to implement.

Since the task is about finding a suitable algorithm, we also expect to see some sort of **unit testing** for the method.

Please use Java 11 or any newer Java version - whatever you are most comfortable with. Also, you’re free to use existing open-source libraries if they help with your task, for example libraries for string alignment, diffing/patching, or others.

What We're Looking For

- Code that is self-explaining and/or that comes with some comments which explain the necessary steps
- Source code management using git (branches, commits, tags, ...)
- If you did something special, then a note stating the reasons behind doing so

This is the place to showcase your skills and you are free to go as deep as you want in any aspect of the task.

Background

Assume that there is some system that takes some written text, looks for things to improve the quality, and outputs an improved version of the text as a replacement suggestion.

For example, when given a long complicated sentence, the system may split it into shorter ones that are easier to understand, remove colloquialisms and exaggerations, etc.:

Input	**Replacement suggested by the system**
`This very loooong text, which includes a lot of useless information, is not easy to understand.`	`This long text includes a lot of information. It is not easy to understand.`

The system is also able to handle HTML-formatted input text. However, the system just removes the HTML elements (tags) from the input, and then processes the resulting plain-text string as before, as in this example:

HTML-formatted input	**Replacement suggested by the system**
`<p>This very <i>loooong</i> text, which includes a <b>lot</b> of useless information, is not easy to understand.</p>`	`This long text includes a lot of information. It is not easy to understand.`

Even with an HTML-formatted input, the output of the system is always a plain-text string. While the actual text may now have a better quality, the original HTML formatting is unfortunately completely lost.

Task

Your task is to write a method that re-inserts the HTML elements from the HTML-formatted input into the plain-text replacement produced by the system at "suitable" text positions. For the previous example, the output could look like this:

HTML-formatted input	Replacement suggested by the system	Output of method
`<p>This very <i>loooong</i> text, which includes a <b>lot</b> of useless information, is not easy to understand.</p>`	`This long text includes a lot of information. It is not easy to understand.`	`<p>This <i>long</i> text includes a <b>lot</b> of information. It is not easy to understand.</p>`

Specifically, we’re asking you to write a method that takes two input strings and returns a string, with the following type:

```
String getHtmlReplacement(final String htmlInput, final String replacement)
```

Given an input string with HTML elements `htmlInput` and a plain-text string `replacement` that does not contain any HTML elements, the method should output a version of `replacement` that contains the HTML elements from `htmlInput.`

The result of your method should be identical to the string `replacement`, but with **all** HTML elements from `htmlInput` added, in the **same order** as they appear in `htmlInput`.

For the task, we assume that HTML elements have the form `<a>` or `</sub>`, i.e. only opening or closing elements with a name that consists exclusively of lower-case English letters, without any HTML attributes, additional spaces, etc.

The problem can admittedly be arbitrarily complex, and it is hard to define what exactly "inserted at suitable text positions" means. There is no correct solution, but here are three things that we would expect:

1. If `htmlInput` doesn’t actually contain any HTML elements, then the result of the method should be identical to `replacement`.

htmlInput	replacement	Output of method
`This is a tyypo.`	`This is a typo.`	`This is a typo.`

2. If removing all HTML elements from `htmlInput` is the same as `replacement` (i.e. the system suggested no change to the plain text of the input), then the output of the method

should be identical to `htmlInput`, i.e. all HTML elements should appear in the same positions again as before.

htmlInput	**replacement**	**Output of method**
`<p>Hello <b>world</b>!</p>`	`Hello world!`	`<p>Hello <b>world</b>!</p>`

3. If `replacement` starts with some prefix text that is identical to the beginning of `htmlInput` after HTML elements have been removed, then these HTML elements should appear in the output at the same positions where they appeared at the beginning of `htmlInput`. Similarly, if the end of `replacement` doesn't differ from the end of `htmlInput` after HTML elements have been removed, then the output should end like `htmlInput`, with HTML elements at the same positions.

htmlInput	**replacement**	**Output of method**
`<b>Good news:</b> This is, fixed!`	`Good news: This is fixed!`	`<b>Good news:</b> This is fixed!</p>`
`I don’t understand <b>this</b>.`	`I do not understand this.`	`I do not understand <b>this</b>.`

Beyond these examples, feel free to explore!

Some hints:

- The problem is conceptually similar to “diffing” two versions of a text (as with “git diff”), or to creating a patch that is then applied onto a third string.
- There is no optimal solution to this problem. We’re more interested in seeing how you approach such a task in general.
- Please do not forget about the requirement that all HTML elements from the input must appear in the output of the method, and in the same order.