

1 Text Segmentation (Dynamic Programming)

1.1 SPLITTABLE

In your programming language of choice, implement a dynamic programming version of the SPLITTABLE algorithm we went over in class from Worksheet 6. Below are examples of an "isWord" helper function that you could use as a placeholder. If you want to use a fancier one that actually checks strings against a dictionary file of English words, you could do that as well.

```
// JAVA
// is the given string a palindrome, with length > 1
public static boolean isWord(String str) {
    StringBuffer sbr = new StringBuffer(str);
    String revstr = sbr.reverse().toString();
    return str.length() > 1 && str.equals(revstr);
}
```

```
# PYTHON
# is the given string a palindrome, with length > 1
def isWord(str):
    return len(str) > 1 and str == str[::-1]
```

1.2 Trace back

Building on the previous implementation, implement a procedure in your language of choice that not only produces true/false, but actually produces the split up list (`ArrayList` or Python list) of valid words. This should work by using the table that the dynamic programming algorithm builds and tracing back through it for a solution.

2 Change Making

Complete Exercise 1, Chapter 3 (page 123) in the textbook. Make sure you read (and follow) the instructions for the "Exercises" section.

Your answer is just expected to be on "paper" and in pseudocode. You do not have to implement in an actual programming language (although you may if you want).