Readability vs. Speed

This lesson discusses a potential problem that arises when you use NumPy library.

Before heading to the next chapter, I would like to warn you about a potential problem you may encounter once you'll have become familiar with NumPy.

It is a very powerful library and you can make wonders with it but, most of the time, this comes at the price of readability. If you don't comment your code at the time of writing, you won't be able to tell what a function is doing after a few weeks (or possibly days).

For example, can you tell what the two functions below are doing? Probably, you can tell for the first one, but unlikely for the second.

```
# More readability, less speed
def function_1(seq, sub):
    return [i for i in range(len(seq) - len(sub) +1) if seq[i:i+len(sub)] == sub]

# More speed, less readability
def function_2(seq, sub):
    target = np.dot(sub, sub)
    candidates = np.where(np.correlate(seq, sub, mode='valid') == target)[0]
    check = candidates[:, np.newaxis] + np.arange(len(sub))
    mask = np.all((np.take(seq, check) == sub), axis=-1)
    return candidates[mask]

seq="Python to NumPy"
sub="NumPy"
print("Function 1:",function_1(seq,sub))
print("Function 2:",function_1(seq,sub))
```

Here in this code, both the functions return the index of the substring in the string. The string is "Python to NumPy" and the substring is "NumPy". The index of the substring in the string is 10. As you may have guessed, the second function is the vectorized-optimized-faster-NumPy version of the first function. It is 10 times faster than the pure Python version, but it is hardly

readable.

Solve this quiz!

1 The code written in Numpy module is vectorized. Is this statement True or False?

COMPLETED 0%

1 of 2

Now that we have a bit introduction to NumPy, let's move on to the next chapter "Anatomy of an Array".