Daily Coding Problem #277

Problem

This problem was asked by Google.

UTF-8 is a character encoding that maps each symbol to one, two, three, or four bytes.

For example, the Euro sign, €, corresponds to the three bytes `11100010 10000010 10101100`. The rules for mapping characters are as follows:

- For a single-byte character, the first bit must be zero.
- For an n-byte character, the first byte starts with n ones and a zero. The other n - 1 bytes all start with `10`.

Visually, this can be represented as follows.

```
 Bytes   |           Byte format
-----------------------------------------------
   1     | 0xxxxxxx
   2     | 110xxxxx 10xxxxxx
   3     | 1110xxxx 10xxxxxx 10xxxxxx
   4     | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

Write a program that takes in an array of integers representing byte values, and returns whether it is a valid UTF-8 encoding.

Solution

Note that for an encoding to be valid, the number of prefix ones in the first byte must

Note that for an encoding to be valid, the number of prefix ones in the first byte must match the number of remaining bytes, and each of those remaining bytes must begin with `10`.

Therefore, we can divide our algorithm into two parts. To start, we check the first element of our input to determine how many remaining bytes there should be, and initialize a counter with that value. Next, we loop through each additional byte. If the byte starts with `10`, we decrement our counter; if not, we can return `False` immediately.

If at the end of our loop, the counter equals zero, we will know our encoding is valid.

```
def valid(data):
    first = data[0]

    if first >> 7 == 0:
        count = 0
    elif first >> 5 == 0b110:
        count = 1
    elif first >> 4 == 0b1110:
        count = 2
    elif first >> 3 == 0b11110:
        count = 3
    else:
        return False

    for byte in data[1:]:
        if byte >> 6 == 0b10:
            count -= 1
        else:
            return False

    return count == 0
```

This algorithm is `O(N)` in the number of bytes, since we are only performing bit shifts and equality checks on each one.

© Daily Coding Problem 2019

Privacy Policy

Terms of Service

Press