393. UTF-8 Validation

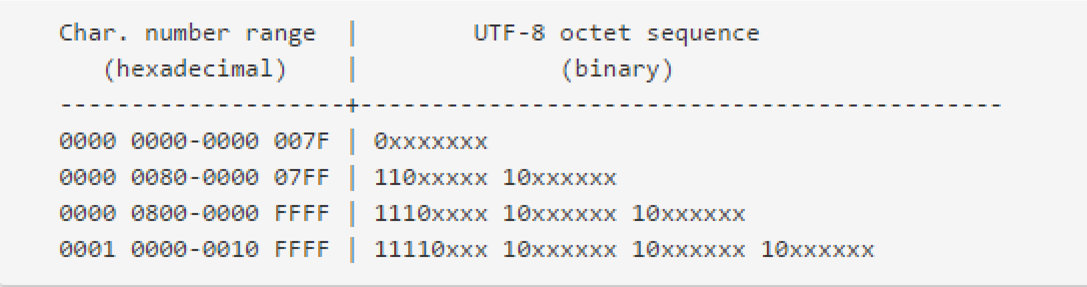
QuestionEditorial Solution
My Submissions

- Total Accepted: **4481**
- Total Submissions: **13004**
- Difficulty: **Medium**
- Contributors: **Admin**

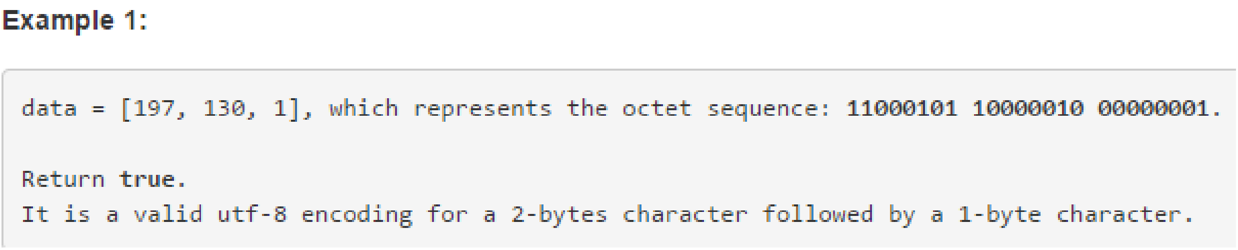
A character in UTF8 can be from 1 to 4 bytes long, subjected to the following rules:

1. For 1-byte character, the first bit is a 0, followed by its unicode code.
2. For n-bytes character, the first n-bits are all one's, the n+1 bit is 0, followed by n-1 bytes with most significant 2 bits being 10.

This is how the UTF-8 encoding would work:



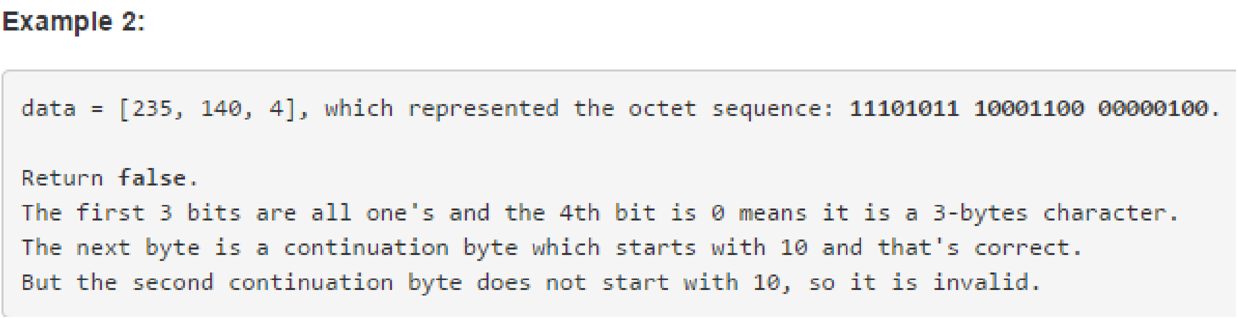
```
   Char. number range  |        UTF-8 octet sequence
      (hexadecimal)    |              (binary)
   --------------------+---------------------------------------------
   0000 0000-0000 007F | 0xxxxxxx
   0000 0080-0000 07FF | 110xxxxx 10xxxxxx
   0000 0800-0000 FFFF | 1110xxxx 10xxxxxx 10xxxxxx
   0001 0000-0010 FFFF | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```



Example 1:

```
data = [197, 130, 1], which represents the octet sequence: 11000101 10000010 00000001.

Return true.
It is a valid utf-8 encoding for a 2-bytes character followed by a 1-byte character.
```



Example 2:

```
data = [235, 140, 4], which represented the octet sequence: 11101011 10001100 00000100.

Return false.
The first 3 bits are all one's and the 4th bit is 0 means it is a 3-bytes character.
The next byte is a continuation byte which starts with 10 and that's correct.
But the second continuation byte does not start with 10, so it is invalid.
```

```
class Solution {
public:
    bool validUtf8(vector<int>& data) {
        int mask1 = 128, mask2 = 192, cnt = 0;
        //(zhewei) mask1 = 0x10000000; mask2 = 0x11000000;
        for(int i=0;i<data.size();i++)
        {
            int cur = data[i];
            if(cnt==0)
            {
                while((cur & mask1)!=0)
                {
                    cur = cur<<1;
                    cnt++;
                }
                //(zhewei) at least 2 for the first number
                // i.e.,110xxxxx 1110xxxx ...
                if(cnt==1) return false;
                // cnt-1 means when the first number has cnt '1' s (i.e.,1110xxxx) in
1110xxxx 10xxxxxx 10xxxxxx
                // its has cnt-1 remaining number (i.e.,10xxxxxx 10xxxxxx)
                cnt = max(0,cnt-1);
            }
            else{
                if((mask2 & data[i]) != mask1) //cur & 0x11000000 != 0x10000000
                return false;
                cnt--;
            }
        }
        return cnt==0;
    }
};
```