# [UTF-8 编码验证](https://leetcode-cn.com/explore/interview/card/bytedance/247/bonus/1037/)

**头条重点**

## 题目

UTF-8 中的一个字符可能的长度为 1 到 4 字节，遵循以下的规则：

* 对于 1 字节的字符，字节的第一位设为0，后面7位为这个符号的unicode码。
* 对于 n 字节的字符 (n > 1)，第一个字节的前 n 位都设为1，第 n+1 位设为0，后面字节的前两位一律设为10。剩下的没有提及的二进制位，全部为这个符号的unicode码。

这是 UTF-8 编码的工作方式：

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

给定一个表示数据的整数数组，返回它是否为有效的 utf-8 编码。

注意：输入是整数数组。只有每个整数的最低 8 个有效位用来存储数据。这意味着每个整数只表示 1 字节的数据。

示例 1:  
  
data = [197, 130, 1], 表示 8 位的序列: 11000101 10000010 00000001.  
  
返回 true 。  
这是有效的 utf-8 编码，为一个2字节字符，跟着一个1字节字符。

## 解题思路

class Solution {  
 public boolean validUtf8(int[] data) {  
 int totalByteCount = 0;  
 for (int item : data) {  
  
 if (totalByteCount == 0) {  
 totalByteCount = totalByteCount(item);  
 if (totalByteCount == -1) {  
 return false;  
 }  
  
 totalByteCount--;  
 continue;  
 }  
  
  
 //10xxxxxx检查  
 if ((item & 0xC0) != 0x80) {  
 return false;  
 }  
  
 totalByteCount--;  
 }  
  
 return totalByteCount == 0;  
 }  
  
  
 private int totalByteCount(int i) {  
 if ((i & 0x80) == 0) {  
 return 1;  
 }  
  
 if ((i & 0xE0) == 0xC0) {  
 return 2;  
 }  
  
 if ((i & 0xF0) == 0xE0) {  
 return 3;  
 }  
  
 if ((i & 0xF8) == 0xF0) {  
 return 4;  
 }  
  
 return -1;  
 }  
}