717. 1-bit and 2-bit Characters

Solved

Easy | Topics | Companies | Hint

We have two special characters:

- The first character can be represented by one bit `0`.
- The second character can be represented by two bits (`10` or `11`).

Given a binary array `bits` that ends with `0`, return `true` if the last character must be a one-bit character.

Example 1:

```
Input: bits = [1,0,0]
Output: true
Explanation: The only way to decode it is two-bit character and one-bit character.
So the last character is one-bit character.
```

Example 2:

```
Input: bits = [1,1,1,0]
Output: false
Explanation: The only way to decode it is two-bit character and two-bit character.
So the last character is not one-bit character.
```

Constraints:

- `1 <= bits.length <= 1000`
- `bits[i]` is either `0` or `1`.

Python:

```
class Solution:
    def isOneBitCharacter(self, bits: List[int]) -> bool:
        n = len(bits)
        if n >= 2 and bits[-2] == 0:
            return True
        i = 0
        while i < n - 1:
            i += bits[i] + 1
```

```
        return i == n - 1
```

JavaScript:

```
const isOneBitCharacter = bits => {
    const n = bits.length;
    let i = 0;
    while(i < n - 1)
        i += bits[i] + 1;
    return i === n - 1;
};
```

Java:

```
class Solution {
    public boolean isOneBitCharacter(int[] bits) {
        int n = bits.length;
        if (n >= 2 && bits[n - 2] == 0) return true;
        int i = 0;
        while(i < n - 1)
            i += bits[i] + 1;
        return i == n - 1;
    }
}
```