

给定一个正整数，检查他是否为交替位二进制数：换句话说，就是他的二进制数相邻的两个位数永不相等。

示例 1:

输入: 5

输出: True

解释:

5的二进制数是: 101

示例 2:

输入: 7

输出: False

解释:

7的二进制数是: 111

示例 3:

输入: 11

输出: False

解释:

11的二进制数是: 1011

示例 4:

输入: 10

输出: True

解释:

10的二进制数是: 1010

思路，该题其实为位运算，若是一个正整数交替，且相邻位不同，那么这个正整数，他右移一位加1与自身，其值刚好位比他大的2的幂倍数。判断一个数是否是2的幂，将该数减1与自身做&按位运算即可，若是为0，就是2的幂倍数。不为0，则不是。

```
class Solution {
public:
    bool hasAlternatingBits(int n) {
        int v=n+1+(n>>1);
        return (v&(v-1))==0;
    }
};
```