

1、给出一个正整数序列，求一个子区间使得这个区间内的数或起来尽可能的大。

或运算指数字按二进制位进行以下运算：

运算规则： $a \oplus b = (a \& \sim b) \cup (\sim a \& b)$

一个序列的子区间指这个序列中连续的一段数字。

牛牛并不关心这个最大值是多少，他只关心所有满足条件的子区间里，最短的子区间长度是多少。

解析：import java.util.Scanner;

```
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int max = 0;
        int n = sc.nextInt();
        sc.nextLine();
        String input = sc.nextLine();
        String[] str = input.trim().split(" ");
        int[] arrs = new int[n];
        for (int i = 0; i < str.length; i++) {
            arrs[i] = Integer.parseInt(str[i]);
            max = max | arrs[i];
        }
        for (int i = 1; i < n + 1; i++) {
            for (int j = 0; j < n - i; j++) {
                int temp = 0;
                for (int k = j; k - j <= i - 1; k++) {
                    temp = temp | arrs[k];
                }
                if (temp == max) {
                    System.out.println(i);
                    return;
                }
            }
        }
    }
}
```