# Советы и секреты

Краткие решения и подсказки

- Задачи

# Задачи



## Разбираем битовую задачку

Сегодня в ежедневной задаче на литкоде попалась интересная штука — нужно найти наименьшее число, которое не меньше заданного N и состоит только из единиц в двоичной записи.

Вам дают число, например, 10 (в двоичной системе это 1010). Нужно найти ближайшее число >= 10, где все биты — единицы. В данном случае это 15 (двоичная 1111).

Первое решение, которое приходит в голову. Можно просто пройтись по всем битам исходного числа и установить каждый в 1:

```
func smallestNumber(n int) int {
    result := 0
    bitPosition := 0

    for n > 0 {
        result |= (1 << bitPosition)  // устанавливаем бит
        bitPosition++
        n >>= 1  // сдвигаем к следующему биту
    }

    return result
}
```

Работает, но есть способ элегантнее. Решение в одну строчку:

Вспомните, как выглядят числа-степени двойки минус один:

$2^1 - 1 = 1$ (двоичная: 1)
$2^2 - 1 = 3$ (двоичная: 11)
$2^3 - 1 = 7$ (двоичная: 111)
$2^4 - 1 = 15$ (двоичная: 1111)

Видите паттерн? Нам нужно найти самый старший бит в исходном числе и взять следующую степень двойки минус один.

Для числа 10 (двоичная 1010) старший бит на позиции 3. Значит, берём $2^4 - 1 = 15$.

```
func smallestNumber(n int) int {
    x := 1
    for x - 1 < n {
        x <<= 1  // умножаем на 2
    }
    return x - 1
}
```

Или через битовые операции с помощью `bits.Len`:

```
import "math/bits"

func smallestNumber(n int) int {
    bitLength := bits.Len(uint(n))  // находим количество бит
    return (1 << bitLength) - 1
}
```

Число с N единицами подряд — это всегда 2^N - 1. Мы находим, сколько битов нужно, чтобы вместить наше число, и берём число с таким количеством единиц.

## FizzBuzz — классика собеседований

Задача проста на вид, но открывает много вопросов на интервью. Вот почему её так любят спрашивать.

Дано число n. Нужно вернуть массив строк, где для каждого числа от 1 до n:

• если кратно и 3 и 5 — «FizzBuzz»
• если кратно 3 — «Fizz»
• если кратно 5 — «Buzz»
• иначе — само число как строку

Решение:

```
func fizzBuzz(n int) []string {
    res := make([]string, n)
```

```
    for i := 1; i <= n; i++ {
        res[i-1] = strconv.Itoa(i)
        if i % 3 == 0 { res[i-1] = "Fizz" }
        if i % 5 == 0 { res[i-1] = "Buzz" }
        if i % 15 == 0 { res[i-1] = "FizzBuzz" }
    }
    return res
}
```

Число 15 — это наименьшее общее кратное чисел 3 и 5. Когда число одновременно делится на 3 и на 5, оно всегда делится на 15.