一、基础位运算

位运算 &（与）

```
    A = 10001001
    B = 10010000
A & B = 10000000
```

位运算 |（或）

```
    A = 10001001
    B = 10010000
A | B = 10011001
```

位运算 ~（非）

```
    A = 10001001
  ~ A = 01110110
```

二、位运算在权限系统中的使用

我们先假定两个前提：

1. 每种权限码都是唯一的
2. 所有权限码的二进制数形式，有且只有一位值为 1，其余全部为 0（比如001、010）

以 Linux 为实例，Linux 的文件权限分为读、写和执行，有字母和数字等多种表现形式：

权限	字母表示	数字表示	二进制
读	r	4	0b100
写	w	2	0b010
执行	x	1	0b001

赋予权限用 |

```
let r = 0b100
let w = 0b010
let x = 0b001
let user = 0b100

// 给用户赋予写的权限
user = user | w
console.log(user)  // 6

//     user = 0b100
//        w = 0b010
// user | w = 0b110
```

校验权限用 &

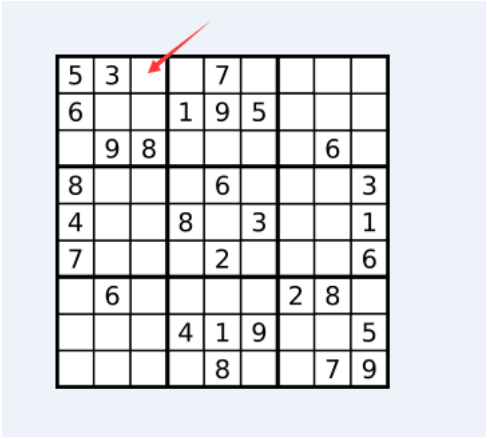
```
let r = 0b100
let w = 0b010
let x = 0b001
let user = 0b110 // 有 r w 两个权限

console.log((user & r) === r) // true  有 r 权限
console.log((user & w) === w) // true  有 w 权限
console.log((user & x) === x) // false 没有 x 权限

//        r = 0b100
//     user = 0b110
// user & r = 0b100

//        x = 0b001
//     user = 0b110
// user & x = 0b000
```

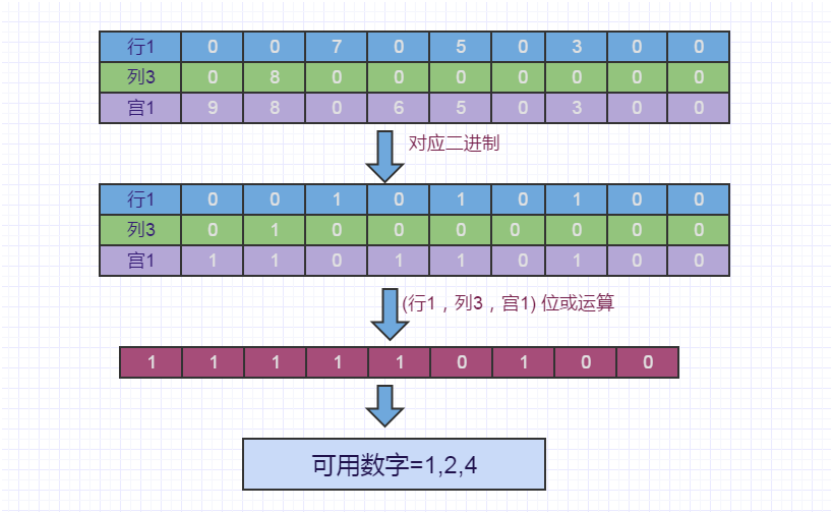
删除权限用 &(~code)

```
let r    = 0b100
let w    = 0b010
let x    = 0b001
let user = 0b110 // 有 r w 两个权限

// 删除 r 权限
user = user & (~r)

//           ~r = 0b011
//        user = 0b110
// user & (~r) = 0b010
```

这里有个数独

问：如何获得当前位置可以填写的数字?





参考：

https://juejin.im/post/5dc36f39e51d4529ed292910