

map 和 parseInt

map 是数组的方法，返回一个执行内部函数后的新函数，不改变原数组。

parseInt(string, radix) 解析一个字符串并返回指定基数的十进制整数，radix 是2-36之间的整数，表示被解析字符串的基数。

parseInt

第一个参数 string 是要被解析的值。如果参数不是一个字符串，则将其转换为字符串(使用 ToString 抽象操作)。字符串开头的空白符将会被忽略。radix 可选，从 2 到 36，表示字符串的基数。例如指定 16 表示被解析值是十六进制数。如果 radix 是 undefined、0 或未指定的，JavaScript 会假定以下情况：

如果输入的 string 以 "0x"或 "0X" (一个0，后面是小写或大写的X) 开头，那么 radix 被假定为 16，字符串的其余部分被当做十六进制数去解析。如果输入的 string 以 "0" (0) 开头，radix 被假定为 8 (八进制) 或 10 (十进制)。具体选择哪一个 radix 取决于实现。ECMAScript 5 澄清了应该使用 10 (十进制)，但不是所有的浏览器都支持。因此，在使用 parseInt 时，一定要指定一个 radix。如果输入的 string 以任何其他值开头，radix 是 10 (十进制)。

要将一个数字转换为特定的 radix 中的字符串字段，请使用 thatNumber.toString(radix)函数。

```
(10).toString(2); // 将 10 转换为 2 进制数，结果是 '1010'
```

['1', '2', '3'].map(parseInt) 返回值是什么？

```
`['1', '2', '3'].map(parseInt)`  
// 等价于  
['1', '2', '3'].map((item, index) => {  
    return parseInt(item, index);  
});  
  
// parseInt('1', 0) // 第二个参数是 0，是特殊情况，如果第一个参数不是以 '0x' 或 '0X'  
// 开头，则当作十进制去解析，返回一个10进制，故返回 1。  
// parseInt('2', 1) // 第二个参数不能大于36，也不能小于2，否则返回 NaN。  
// parseInt('3', 2) // 将 3 当作 2 进制解析，2 进制只能包含 0 或 1，所以解析失败，返回  
// NaN。
```

根据上面的规则返回 [1, NaN, NaN]。

解题方法，将问题拆解到最细粒度再分析。