

练习题 生成随机矩阵（连连看）

假如要利用函数随机生成一个二维数组，数组要求长度为列数cols*行数rows，指定数组中的数据由1到n的成对数据填充(并打乱顺序)，比如

```
cols = 5
rows = 6
n = 5
```

那么总计数组长度应该是30，使用 1-5的成对数据填充后结果可能是这样

```
3 1 2 1 2
1 4 3 5 1
2 1 1 5 1
3 2 3 1 1
5 4 1 1 5
4 1 3 4 3
```

该函数应该怎样生成？

解答：

结果需要是一个二维数组吗？其实一维数组就可以，二维数组只是为了让我们看起来好像更像矩阵一点

如果需求是生成一个3x4的矩阵，意思就是3列4行，他的一维数组形式是：

```
[
    2,4,2,3,4,1,4,3,4,1,1,1
]
```

只要我们把格式稍微调整一下，写成这样就很容易明白这个矩阵是什么样的了

```
[
    2,4,2,
    3,4,1,
    4,3,4,
    1,1,1
]
```

那么剩下的问题就是如何生成这个一维数组，其实前面的练习这些工作都做过了，无非分为两步

第一步：填充随机数（1-n范围内的）

第二步：打乱顺序

而第一步中又有一个要求是数字必须成对出现，所以填充过程要循环的次数是

`(rows*cols)/2`

看一下赵聪的实现：

第一步：填充

```
var procArray = [...Array((col * row) / 2)].map(e => parseInt(Math.random() * n, 10) + 1)

procArray = procArray.concat(procArray)
```

循环col*row/2次 填充随机数

然后把得到的数组自己和自己相连一下，等于翻了倍

第二步：乱序

```
procArray.sort((a, b) => Math.random() > 0.5 ? 1 : -1)
```

如果要得到一个二维数组，就剩下再分割一下这个一维数组的数据而已了。

所以，如果有人没有实现的这么简练的话，其实根本原因不是语法熟悉与否的问题，而是没有把解决问题的思路想清楚。

最后要注意

假如获取一个3x4的矩阵，结果如下

```
▼ (4) [Array(3), Array(3), Array(3), Array(3)] ⓘ
  ▶ 0: (3) [2, 1, 1]
  ▶ 1: (3) [2, 1, 1]
  ▶ 2: (3) [3, 2, 1]
  ▶ 3: (3) [2, 1, 3]
    length: 4
  ▶ __proto__: Array(0)
```

要得到箭头所指处的数字，其坐标是(2, 1)但取数组的下标应该为[1][2]，是倒过来的