

384. Shuffle an Array

【宫水三叶】洗牌算法模板题[宫水三叶](#) 洗牌算法

共有 n 个不同的数，根据每个位置能够选择什么数，共有 $n!$ 种组合。

题目要求每次调用 shuffle 时等概率返回某个方案，或者说每个元素都够等概率出现在每个位置中。我们可以使用 Knuth 洗牌算法，在 $O(n)$ 复杂度内等概率返回某个方案。

具体的，我们从前往后尝试填充 $[0, n - 1]$ 该填入什么数时，通过随机当前下标与（剩余的）哪个下标进行值交换来实现。

对于下标 x 而言，我们从 $[x, n-1]$ 中随机出一个位置与 x 进行值交换，当所有位置都进行这样的处理后，我们便得到了一个公平的洗牌方案。

对于下标为 0 位置，从 $[0, n-1]$ 随机一个位置进行交换，共有 n 种选择；下标为 1 的位置，从 $[1, n-1]$ 随机一个位置进行交换，共有 $n-1$ 种选择 ...

代码：

```
class Solution {
    int[] nums;
    int n;
    Random random = new Random();
    public Solution(int[] _nums) {
        nums = _nums;
        n = nums.length;
    }
    public int[] reset() {
        return nums;
    }
    public int[] shuffle() {
        int[] ans = nums.clone();
        for (int i = 0; i < n; i++) {
            swap(ans, i, i + random.nextInt(n - i));
        }
        return ans;
    }
    void swap(int[] arr, int i, int j) {
        int c = arr[i];
        arr[i] = arr[j];
        arr[j] = c;
    }
}
```

时间复杂度： $O(n)$ 空间复杂度： $O(n)$