

leetcode 380

在O(1)的时间内对一个数组进行增删查。

分析：

1. 因为插入和删除都需要O(1)的时间，所以考虑unordered_map结构；
2. 因为需要随机获取，且时间复杂度要求O(1),所以采用vector<int>格式。
3. 在使用vector<int>时，为保证删除的时间复杂度，在unordered_map的key为元素值，value为元素在vector<int>中的索引。
4. 删除时，交换vector<int>的末尾元素与index值，弹出back即可。

```
class RandomizedSet {
public:
    RandomizedSet() {
    }

    bool insert(int val) {
        if (map_.count(val)) {
            return false;
        }
        val_.push_back(val);
        map_[val] = val_.size() - 1;
        return true;
    }

    bool remove(int val) {
        if (!map_.count(val)) {
            return false;
        }

        int index = map_[val];
        int key = val_.back();
        std::swap(val_[index], val_.back());
        map_[key] = index;
        map_.erase(val);
        val_.pop_back();

        return true;
    }

    int getRandom() {
        if (val_.size()) {
            return val_[rand() % val_.size()];
        }
        return -1;
    }

private:
```

```
std::vector<int> val_; // 存储元素
std::unordered_map<int, int> map_; // 存储元素与索引
};
```