# 题目

RandomizedCollection 是一种包含数字集合(可能是重复的)的数据结构。它应该支持插入和删除特定元素，以及删除随机元素。

实现 RandomizedCollection 类:

RandomizedCollection()初始化空的 RandomizedCollection 对象。

bool insert(int val) 将一个 val 项插入到集合中，即使该项已经存在。如果该项不存在，则返回 true ，否则返回 false 。

bool remove(int val) 如果存在，从集合中移除一个 val 项。如果该项存在，则返回 true ，否则返回 false 。注意，如果 val 在集合中出现多次，我们只删除其中一个。

int getRandom() 从当前的多个元素集合中返回一个随机元素。每个元素被返回的概率与集合中包含的相同值的数量 线性相关 。

您必须实现类的函数，使每个函数的 平均 时间复杂度为 O(1) 。

注意：生成测试用例时，只有在 RandomizedCollection 中 至少有一项 时，才会调用 getRandom 。

示例 1:

输入

["RandomizedCollection", "insert", "insert", "insert", "getRandom", "remove", "getRandom"]

[[], [1], [1], [2], [], [1], []]

输出

[null, true, false, true, 2, true, 1]

解释

RandomizedCollection collection = new RandomizedCollection();// 初始化一个空的集合。

collection.insert(1); // 返回 true，因为集合不包含 1。

// 将 1 插入到集合中。

collection.insert(1); // 返回 false，因为集合包含 1。

// 将另一个 1 插入到集合中。集合现在包含 [1,1]。

collection.insert(2); // 返回 true，因为集合不包含 2。

// 将 2 插入到集合中。集合现在包含 [1,1,2]。

collection.getRandom(); // getRandom 应当:

// 有 2/3 的概率返回 1,

// 1/3 的概率返回 2。

collection.remove(1); // 返回 true，因为集合包含 1。

// 从集合中移除 1。集合现在包含 [1,2]。

collection.getRandom(); // getRandom 应该返回 1 或 2，两者的可能性相同。

提示:

-231 <= val <= 231 - 1

insert, remove 和 getRandom 最多 总共 被调用2 \* 105 次

当调用getRandom时，数据结构中 至少有一个元素

# 分析