# 题目

给定一个可能含有重复元素的整数数组，要求随机输出给定的数字的索引。 您可以假设给定的数字一定存在于数组中。

注意：

数组大小可能非常大。使用太多额外空间的解决方案将不会通过测试。

示例：

int[] nums = new int[] {1,2,3,3,3};

Solution solution = new Solution(nums);

// pick(3)应该返回索引2,3或者4。每个索引的返回概率应该相等。

solution.pick(3);

// pick(1) 应该返回0。因为只有nums[0]等于1。

solution.pick(1);

# 分析

## 方法一：哈希表

## 方法二：蓄水池抽样

**思路：**

水塘抽样，参见知乎https://zhuanlan.zhihu.com/p/29178293

对于需要随机的数字做计数i，i=1，则概率就是1

i=2，概率就是1/2

i=3, 概率就是1/3

总结就是按照当前计数i，只取1/i的概率选择该结果，那么最后对于所有1-都是是平均1/i的概率。

**代码：**

class Solution {

private:

vector<int> nums;

int n;

public:

Solution(vector<int>& nums) : nums(nums) {

n = nums.size();

}

int pick(int target) {

// 计数器，默认是1

int cnt = 1;

int res = 0;

for (int i = 0; i < n; ++i)

{

int num = nums[i];

if (num == target)

{

// 发现预期的数字，则开始一次随机数，如果正好是1/i概率则更新结果

if ((rand() % cnt + 1) == 1)

{

res = i;

}

++cnt;

}

}

return res;

}

};

另一种写法：

class Solution {

private:

    vector<int> nums;

    int n;

public:

    Solution(vector<int>& nums): nums(nums) {

        n = nums.size();

    }

    int pick(int target) {

        int count = 1;

        int ret = 0;

        for(int i=0;i<n;i++)

        {

            int num = nums[i];

            if(num == target)

            {

                if(rand()%count==0)

                {

                    ret = i;

                }

                count++;

            }

        }

        return ret;

    }

};

/\*\*

 \* Your Solution object will be instantiated and called as such:

 \* Solution\* obj = new Solution(nums);

 \* int param\_1 = obj->pick(target);

 \*/

**思路：**

蓄水池抽样算法，大数据流中的随机抽样问题，从未知大小并且数据只能访问一次的数据流随机选取k个数据，保证每个数据被抽取到的概率相等。

k步之前，被选中概率为1，k步走后，如果第i步被选中的概率是k/cnt，则最终所有数据被选中概率是k/n。

用(rand()%cnt) ==0表示1/cnt的概率第i个数字被选中

固定cur，如果\*cur是target，对cnt抽样

**代码：**

class Solution {

public:

vector<int>::iterator beg,end;

Solution(vector<int>& nums) {

beg=nums.begin();

end=nums.end();

}

int pick(int target) {

vector<int>::iterator cur=beg;

int cnt=1;

int pool=0;

int i=0;

while(cur!=end){

if(\*cur==target){

if(rand()%cnt==0)pool=i;

cnt++;

}

i++,cur++;

}

return pool;

}

};