哈希表(Hash table)

① 哈希表是根据关键码的值而直接进行访问的数据结构.

② 数组就是一张哈希表.

③ 哈希表用来快速判断一个元素是否出现集合里.

④ 哈希函数:

① 通过特定的编码方式将名字等转化为数值与数组的索引相对应.

② 如果取的数值 > tablesize, 对转换的数值在做一个取模操作.

⑤ 哈希碰撞:

避免不同的名字转换后的数值相同. { 拉链连法. / 线性探测法.

1. 拉链连法.

不同的名字在索引处发生冲突, 发生冲突的元素被存储在链表中. 再通过索引找到小王和小李.

选择适当的哈希表的大小.

2. 线性探测法:

① 一定要保证 tablesize > datasize, 我们需要依靠哈希表中的空位来解决碰撞问题.

② 如果发生冲突, 先放1, 其余空余位置放2

⑥ 常见的三种哈希结构: 数组. set(集合), map(映射)

需要快速判断一个元素是否出现集合里的时候, 考虑哈希法.

牺牲空间换取时间.

242. 有效的字母异位. → s和t中每个字符出现次数相同.

给定两个字符串s和t. 编写一个函数来判断t是否是s的字母异位词.

input: s="anagram" t="nagaram" ⇒ true

input: s="rat" t="car" ⇒ False

伪代码.

1. 判断两个字符串长度是否一样

```
for i in s:
    if 判断i在s中的次数与在t中是否一样:
        return False
return true.
return False.
```

2. 卡哥思路:

定义一个哈希表来记录字母的次数在Python中字典就是.

使用 s="aec" t="eae" 举例

索引: 0 1 2 3 4 5 6

元素: [ ][ ][ ][ ][ ][ ][ ]

① 定义一个新数组记录次数, 数组大小为26

② 使用 s[i]-'a' 来获得相对索引. → Python中使用ord来取ASCII码

③ 对于s字符串, 遍历并在新数组中记录count+1

④ 对于t字符串, 遍历并在新数组中记录count-1

⑤ 最后检查新数组每个元素是否为0 → 为0→true; 不为0→False

349. 两个数组的交集

给定两个数组，编写一个函数来计算它们的交集。

nums1 = [1,2,2,1]
nums2 = [2,2] ⇒ [2]

思路1：

record = { } → set( )

```
for i in nums1:
    if i in nums2:
        record[i] = 1 → record.add(i)
return [key for key, value in record.items() if value != 0]
```

1. 使用字典记录，如果该值在两个数组中都出现，则记录在字典中
2. 最终遍历如果值 != 0 则返回对应键。

思路2：

使用数组 总大小 [0, 1000] 1001个数 → 根据索引值来创建哈希表

202. 快乐数

判断一个数是不是快乐数。

对于一个整数n，每一次将该数替换为它每个位置上的数字的平方和。重复这个过程直到这个数变为1，也可能是无限循环，如果为1，就是快乐数

代码：

```
while n != 1 and n != 4:
```

```
n=sum([int(i)**2 for i in str(n)])
return n==1.
```

1. 两数之和

给定一个整数数组 nums 和一个目标值 target，在数组中找出和为目标值的两个整数，并返回他们的数组下标.

```
records = {}
for index, value in enumerate(nums):
        if target-value in records:
            return [records[target-value], index]
        records[value] = index
```