

# Hash table

#G面试准备

## Longest Substring with At Most K Distinct Characters

Longest Substring with At Most K Distinct Characters

- 和LRU是一个道理
- hashmap存char和它最后出现的index
- hashmap的size如果超过k 就要删除index最老的那个char 并从那个char后面开始计长度
- 维护一个全局max
- 注意low = Collections.min(dict.values());
  - dict.values()返回所有的values Collections虚类型
  - 用Collections.min来找其中的最小值
- 循环每一次都要比max和i - low + 1:
  - 如果dict的size没有超过k low不会变
  - 超过k了 low要更新到被删除的char的后一位
  - 记得加一

## LRU Cache

LRU Cache - LeetCode

- 双向链表实现
- LRU初始化的时候会定义一个head和一个tail node 都不存东西 都是空的 所有的东西夹在这中间
- 要做一个moveToTail函数
- put先用到了get来判断是否存在 存在的话可以直接用get来改优先顺序 并返回结果
- get也要改顺序 方法就是从中间删节点 改两边节点的指针 然后把当前节点moveToTail
- put如果到了capacity要先删head.next这个node 并且要记得删hashmap里对应的kv pair
- 插入新节点的时候记得要加入hashmap 用来根据key来找到所需要的链表里的node value存在node里

## 49. Group Anagrams

Group Anagrams - LeetCode

- 面经题
- anagram: 如果从第一个string中字母变换一下顺序能变成第二个string 就是anagram
- valid anagram里就是用hashmap去存第一个词里每个字的出现次数 然后和第二个词的count去比对

- 这题可以把每个词都sort了然后存hashmap里 如果两个词sort之后是一样的string就说明互相为anagram

## 128. Longest Consecutive Sequence

### Longest Consecutive Sequence - LeetCode

First turn the input into a set of numbers. That takes  $O(n)$  and then we can ask in  $O(1)$  whether we have a certain number.

Then go through the numbers. If the number  $x$  is the start of a streak (i.e.,  $x-1$  is not in the set), then test  $y = x+1, x+2, x+3, \dots$  and stop at the first number  $y$  not in the set. The length of the streak is then simply  $y-x$  and we update our global best with that. Since we check each streak only once, this is overall  $O(n)$ . This ran in 44 ms on the OJ, one of the fastest Python submissions.

```
def longestConsecutive(self, nums):
    nums = set(nums)
    best = 0
    for x in nums:
        if x - 1 not in nums:
            y = x + 1
            while y in nums:
                y += 1
            best = max(best, y - x)
    return best
```

- 看起来是 $O(n)$  其实非常厉害
- `if x - 1 not in nums` 这一句会从连续数列的起点开始 如果是在数列中间的数则不会进入if下面的部分
- 所以其实每个数字都只走一次