

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