

1. Most Letter/Shopping Pattern

lc对应链接：

<https://leetcode.com/discuss/interview-question/381353/Affirm-or-Phone-Screen-or-Letters-appearing-most-number-of-words>

<https://leetcode.com/discuss/interview-question/1882306/Affirm-or-Phone-or-Max-Company-Count>

<https://www.1point3acres.com/bbs/thread-795276-1-1.html>

2. Shortest Substring

followup 1: 优化一下substr(), 就是用字符串长度来做循环

followup 2: 不一定存在unique, 输出最短的substring

<https://www.1point3acres.com/bbs/thread-847354-1-1.html>

3. Insert Delete GetRandom O(1)

<https://www.1point3acres.com/bbs/thread-841644-1-1.html>

可以使用Java built in的data structure, 实现get(key), get_random_val(), put(key, value), delete(), 除了delete其他都应该是constant time

```
{
"a": "apple",
"b": "banana",
"c": "peach",
"d": "pear",
}
```

等概率返回1/4 "a".val, 1/4 "b".val, 1/4 "c".val, 1/4 "d".val.

实现起来挺简单的, 就是存一个HashMap和一个List of keys, 每次get_random的时候可以random.nextInt(keys.size()), return对应的value

follow up是input可能有different value, 但需要等概率返回value

比如

```
{
"a": "apple",
"b": "banana",
"c": "apple",
"d": "apple",
}
```

unique value是apple和banana, 所以应该1/2 apple, 1/2 banana

需要改的地方是maintain一个List of value还有额外一个HashMap存每个value对应的count, 在delete的时候减为0时从value list里删除

<https://www.1point3acres.com/bbs/thread-814471-1-1.html>

LeetCode 三巴伊的变形, 设计一个data structure满足以下要求

```
public void put(int key, int val);
```

```

public int get(int key);
public void remove(int key);
public int getRandomValue();

```

getRandomValue 按相同概率返回其中任意一个value
Follow up是允许不同的key有相同的value
比如 (1, 100), (2, 100), (3, 200)
要求设计两个方法

```

public int getRandomValue();
public int getUniqRandomValue();

```

getRandomValue 需要按frequency来, duplicate越多, 概率越大
getUniqRandomValue不看, 不同的value概率相同, 不论出现多少次duplicate
e.g.
getRandomValue :
100 -> 2/3
200 -> 1/3
getUniqRandomValue :
100 -> 1/2
200 -> 1/2

设计一个randomized dictionary , get, put, delete都和正常dictionary一样, getRandom 随机返回某一个key/value pair的value (example: put(0,1), put(1,1), put(2,2) getRandom 2/3概率返回1, 1/3概率返回2)
follow up是getRandom改成随机返回dictionary的values (example: put(0,1), put(1,1), put(2,2) getRandom 50%返回1, 50%返回2)
另外最开始要求是delete允许O(n) 其他要求O(1) , optimize要求都用O(1)

<https://www.1point3acres.com/bbs/thread-729448-1-1.html>

4. Insert Delete GetRandom O(1) duplicate

[https://leetcode.com/discuss/interview-question/1591333/Affirm-or-phone-interview-or-Insert-Delete-GetRandom-O\(1\)](https://leetcode.com/discuss/interview-question/1591333/Affirm-or-phone-interview-or-Insert-Delete-GetRandom-O(1))

<https://leetcode.com/discuss/interview-question/1894384/AFFIRM-or-OA-ROUND-2or-Random-Access-With-Duplicates>

5. Design a hit count

<https://www.1point3acres.com/bbs/thread-854911-1-1.html>

写两个API process_loan 和 getLoan()(名字不记得也不重要) . 第二个API 要求返回1小时内的amount.

LZ最近在刷题期, 而且由于这又是第一面脑子一直没转过弯, 一直当刷题。。。但其实对面很想听design的东西。

follow up 我答的就是优化用timestmap做key。把popup one hour 的工作丢到process_loan 这样就可以只maintain 1hour timeframe 的数据。这样就都是constant 了。但是当时脑子没转过弯这段虽然答出来了但是没答好估计挂了。

coding 是之前有人分享過的題, 但當時沒有很懂題意:

process_loan(amount) -> None

amount is a dollar amount

get_loan_volume() -> Decimal

the total \$\$ amount of loans processed in the last 1 hour

一開始先直覺地把(time, amount) 存在一個list, get loan 的時候再用binary search 找出index, then sum up array[index:], 這裡要求runnable code

面試官接著要求優化兩個function 都要O(1), 突然想到題目只在乎sum, 所以可以用一個variable 紀錄過去一個小時的sum就好, 再用一個bucket array 紀錄每一秒內的sum

每當移動到下一個bucket的時候, 需要先把total sum - bucket sum, 然後把current bucket sum 歸零, 再把當前的amount 加到total sum 和 bucket sum裡面。

跟面試官解釋過code 之後, 他就說ok滿意了

让你实现两个支付接口，第一个接口是记录下每笔存入的金额与时间，接口二是返回过去60分钟所有存入的钱。 要求，时间颗粒度（granularity）到秒；put, get调用频率很高，需要常数时间复杂度。

def putTransaction(amount, timestamp):

def getTotalTransactionInLastOneHour():

Solution 1: Circle Array

Solution 2: FIFO queue

<https://www.1point3acres.com/bbs/thread-715404-1-1.html>

6. Pop Up node

附一个pop up, 扫一遍 不用加parent

1. 从root到POPUP的path 上所有点 hidden=false

2. 从root到POPUP的path上所有点的sibling hidden=true

3. POPUP 的 sibling hidden=true

4. 其他的点不变

```
bool dfs(DomNode* root) {
    if (!root) { return false; }
    if ("POPUP" == root->id) {
        root->hidden = false;
        return true;
    }
    bool has_popup = false;
    unordered_map<DomNode*, bool> node_has_popup;
    for (auto child : root->children) {
        node_has_popup[child] = dfs(child);
        has_popup = has_popup or node_has_popup[child];
    }
    if (has_popup) {
```

```

        for (auto child : root->children) {
            if (node_has_popup[child]) {
                child->hidden = false;
            }
            else {
                child->hidden = true;
            }
        }
    }
    return has_popup;
}
void open_popup(DomNode* root){
    if (!root) {
        return;
    }
    bool has_pop = dfs(root);
    if (has_pop) {
        root->hidden = false;
    }
    return;
}

```

7. card game <https://leetcode.com/discuss/int ... Affirm-Phone-screen>

<https://www.1point3acres.com/bbs/thread-774866-1-1.html>

```

// Part 1:
// this is a two player card game
// the game starts with a deck of 52 cards represented as unique integers [1...52]
// the cards are randomly shuffled and then dealt out to both players evenly
// on each turn:
// both players turn over their top-most card
// the player with the higher valued card takes the cards and puts them in their scoring pile
// (scoring 1 point per card)
// this continues until all the players have no cards left
// the player with the highest score wins
// if they have the same number of cards in their win pile, tiebreaker goes to the player with
the highest card in their win pile

```

8. Decision tree

<https://www.1point3acres.com/bbs/thread-877599-1-1.html>

coding: parse and store decision tree,要求写3个method 1. 建 decision tree (left or right child 可以一步一步call 然后开始建) 2.evaluate decision (给所有decision 的 value , 做判断 , 是 true

/ false) 3. store decision tree (大概意识是如何把这个建好的decision tree存好，发给别的host 然后做decision)

<https://www.1point3acres.com/bbs/thread-770612-1-1.html>

<https://www.1point3acres.com/bbs/thread-568749-1-1.html>

<https://www.1point3acres.com/bbs/thread-568416-1-1.html>

9. filter

<https://www.1point3acres.com/bbs/thread-726848-1-1.html>

给的是一个file path, 让你读这个文件，然后parse, 最后filter 结果，得到他们想要的东西

color date number

green 2001/02/23 8

purple 2006/05/11 1

white 2019/02/17 200

```
sheet = SpreadSheet("a.txt");
```

```
sheet.filter(['color', '=', 'green'])
```

=>

```
[  
  ['green', 2001/02/03, 8]
```

```
]
```

注意，颜色可能有duplicate.

希望大家可以加点米，谢谢，我自己都看不到自己的帖子。。。