Warm-up

Problem 1. Consider a hash table of size $N > 1$, and the hash function such that $h(k) = k \bmod 2$ for every k. We insert a dataset S of size $n < N$. After that, what is the typical running times of GET for chaining and open addressing (as a function of n)?

Problem 2. Suppose you are given a hash function h mapping 10-digit integers to integers in $\{1, 2, \ldots, 10000\}$. Show that there is some dataset S of size $1,000,000$ such that all keys of S are hashed to the *same* value.

Problem 3. Work out the details of implementing cycle detection in cuckoo hashing based on the number of iterations of the eviction sequence.

Problem 4. Work out the details of implementing cycle detection in cuckoo hashing based on keeping a flag for each entry.

Problem solving

Problem 5. Design a sorted hash table data structure that performs the usual operations of a hash table with the additional requirement that when we iterate over the items, we do so in the order in which they were inserted into the hash table. Iterating over the items should take $O(n)$ time where n is the number of items stored in the hash table. Your data structure should only add $O(1)$ time to the standard put, get, and delete operations.

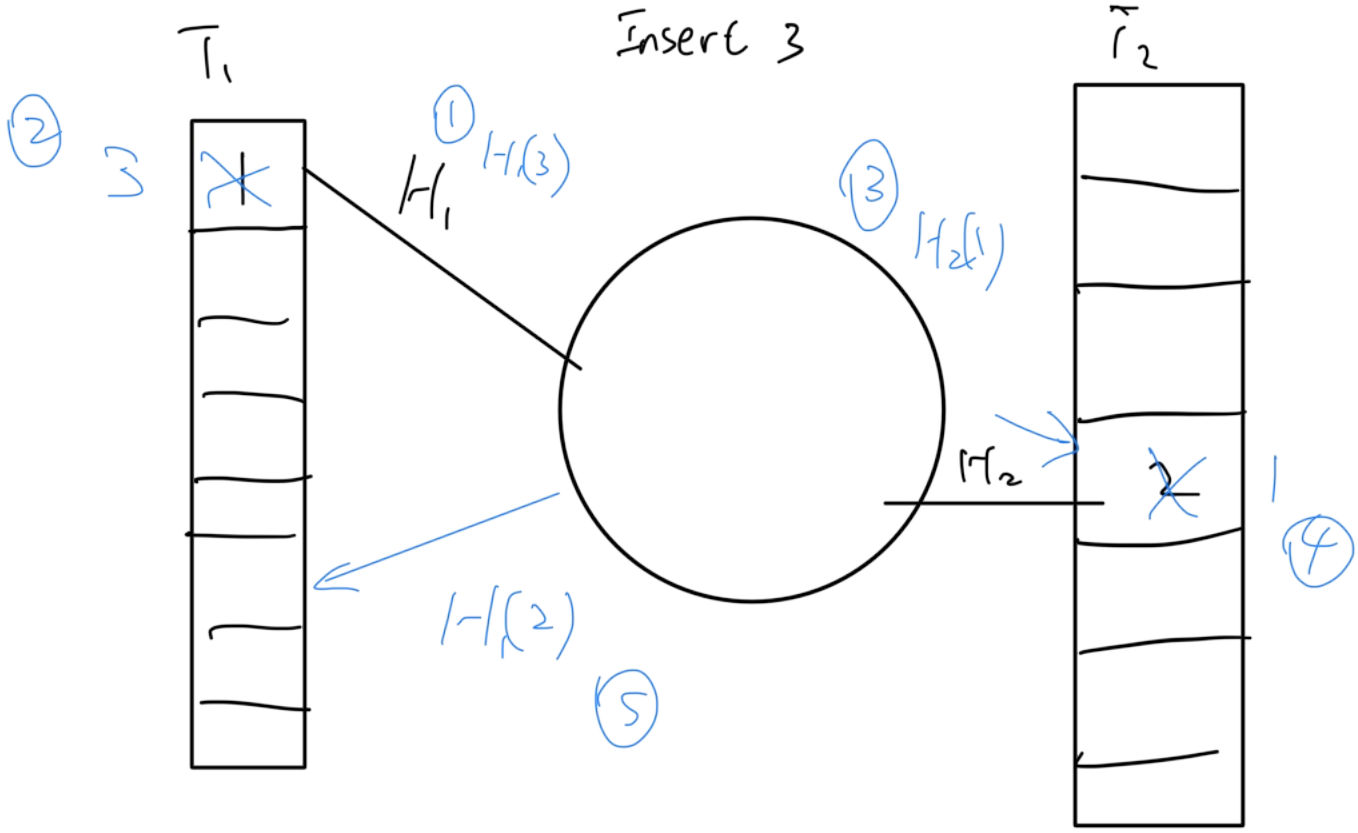
Problem 6. Given an array with n integers, design an algorithm for finding a value that is most frequent in the array. Your algorithm should run in $O(n)$ expected time.

Problem 7. A multimap is a data structure that allows for multiple values to be associated with the same key. The method GET(k) should return all the values associated with key k. Describe an implementation where this method runs in $O(1 + s)$ expected time, where s is the number of values associated with k.

Problem 8. Suppose that you have a group of n people and you would like to know if there are two people that share a birthday. Design an $O(1)$ time algorithm that given the information about the n people's birthdays, finds a pair that shares a birthday, or reports that no such pair exists.

Problem 9. In computational linguistics, texts (such as a book or an article) are modelled as a sequence of words. A k-gram is a sequence of k consecutive words. A common task in language modelling requires that we compute the frequency of all k-grams that appear in the text.

Given a text with n words, design an $O(n)$ expected time algorithm that computes the frequency of all k-grams that appear at least once in the text.

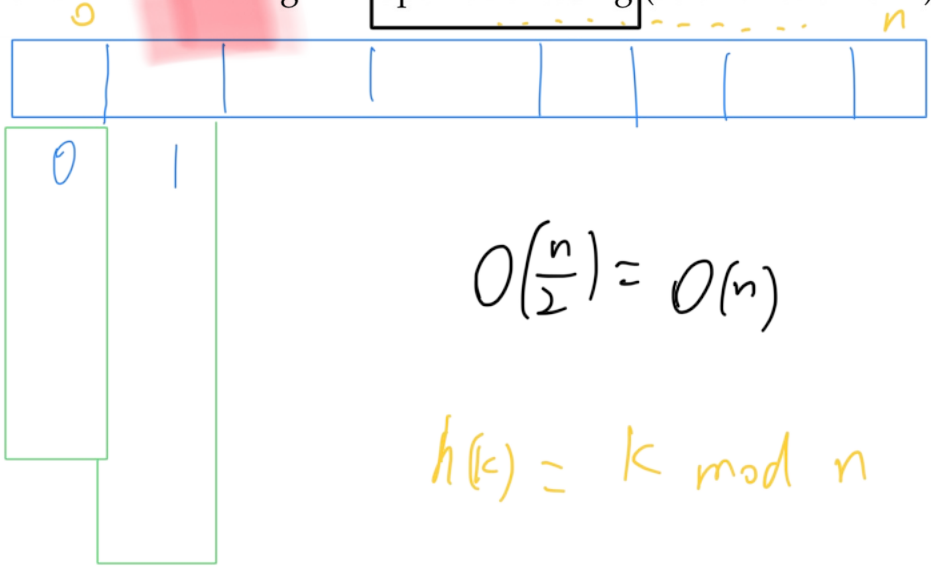


Explaination about cuckoo

对于 linear probing

Problem 1. Consider a hash table of size $N > 1$, and the hash function such that $h(k) = k \bmod 2$ for every k. We insert a dataset S of size $n < N$. After that, what is the typical running times of GET for chaining and open addressing (as a function of n)?

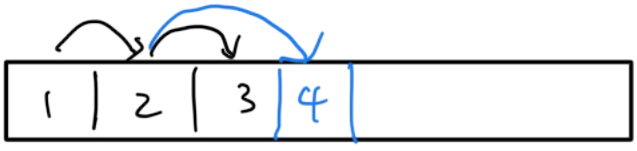
①



$O(\frac{n}{2}) = O(n)$

$h(k) = k \bmod n$ 才能平均分布 space

②



$O(n \cdot 1) = O(n)$

Problem 2. Suppose you are given a hash function h mapping 10-digit integers to integers in $\{1, 2, \ldots, 10000\}$. Show that there is some dataset S of size $1,000,000$ such that all keys of S are hashed to the *same* value.

- Hash Table : $\{1, \ldots, 10{,}000\}$

  size is $10{,}000 = 10^4$

- 10 digits integer : $[10^9, 10^{10}-1]$ 考虑前导0; 9×10^9 num

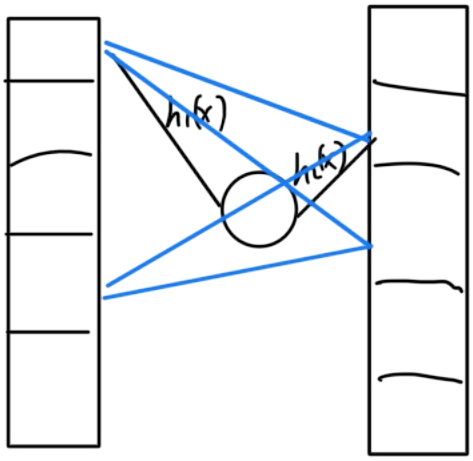
  we use this → $[0, 10^{10}-1]$ 不考虑前导0 10^{10} num

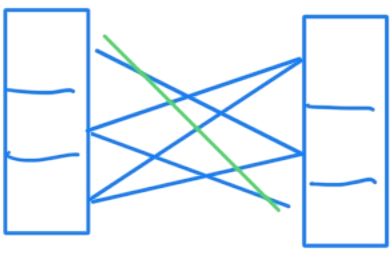
  because aswer said so

  $\frac{10^{10}}{10^4} = 10^6$; so there are 10^6 number map to the same position

Problem 3. Work out the details of implementing cycle detection in cuckoo hashing based on the number of iterations of the eviction sequence.

Example of cycle





—— : means eviction

There are 2N entry. So in the worst case, there can be 2N shifts to move all entries to their alternative position and another 2N to move everything back to their initial position. Then we conduct it is a cycle. we can check it by adding a counter which increase when we do eviction

Problem 4. Work out the details of implementing cycle detection in cuckoo hashing based on keeping a flag for each entry.

- 所有 entry 都有 flag. 且初使设为 False; 其中 True 表示当前检查过该 entry了
- Assume $i = h_1(x)$ $j = h_2(x)$

- Test_chain(x, i): 检查从 i 开始的 eviction chain

```
if i is empty:
    return True
else:
    if( i is not empty && i.flagged == True):
        return False // 表示有 cycle
    else: // 即 i is not empty && i.flagged == False
        // 假设 i 现在存着 y
        replace y by x
        i.flagged = True
        return Test-Chain(y, k) where k is the alternative entry
                                for y
```

- We need use Test chain for both i and j when we insert x. Because the eviction chain might be different for both case. If both get False, we conclude eviction cycle
- 当我们要再插入新的 element, 我们需要把所有的 flagged set to false

Problem 5. Design a sorted hash table data structure that performs the usual operations of a hash table with the additional requirement that when we iterate over the items, we do so in the order in which they were inserted into the hash table. Iterating over the items should take $O(n)$ time where n is the number of items stored in the hash table. Your data structure should only add $O(1)$ time to the standard put, get, and delete operations.

我们要 iterate 这些 items, order 是 which they were inserted into the hash table.

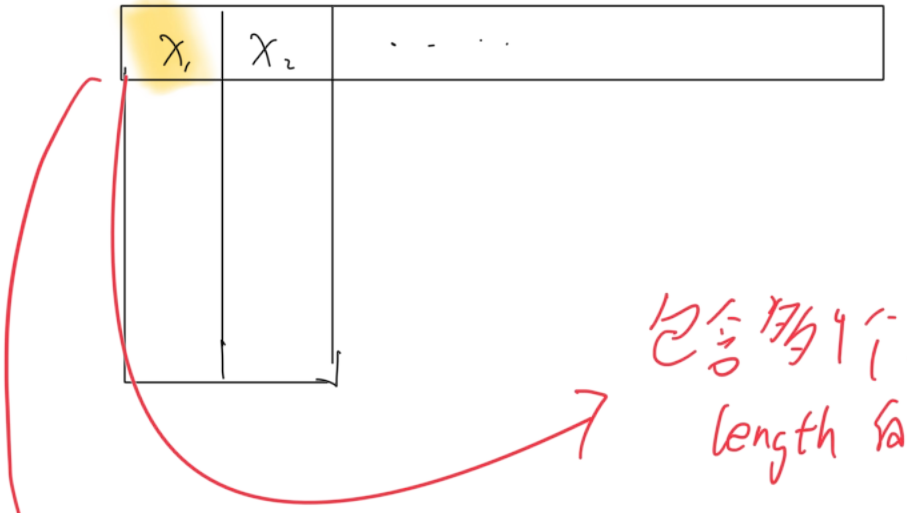
- 用另外一个 linked list 记录顺序?
- 注意, Hash Table 不包含 iteration 的功能,
  所以我们要再用一个 doubly linked list 来记录插入顺序
- Hash Table node structure:
  (key, pointer to node in Doubly Linked list)

DLL
- node.value // 正常 dict 中的 value
- node.next
- node.last

Problem 6. Given an array with n integers, design an algorithm for finding a value that is most frequent in the array. Your algorithm should run in $O(n)$ expected time.

$[x_1, \cdots, x_n]$

$h(x_i)$, same x_i will store in the same place in Hash Table



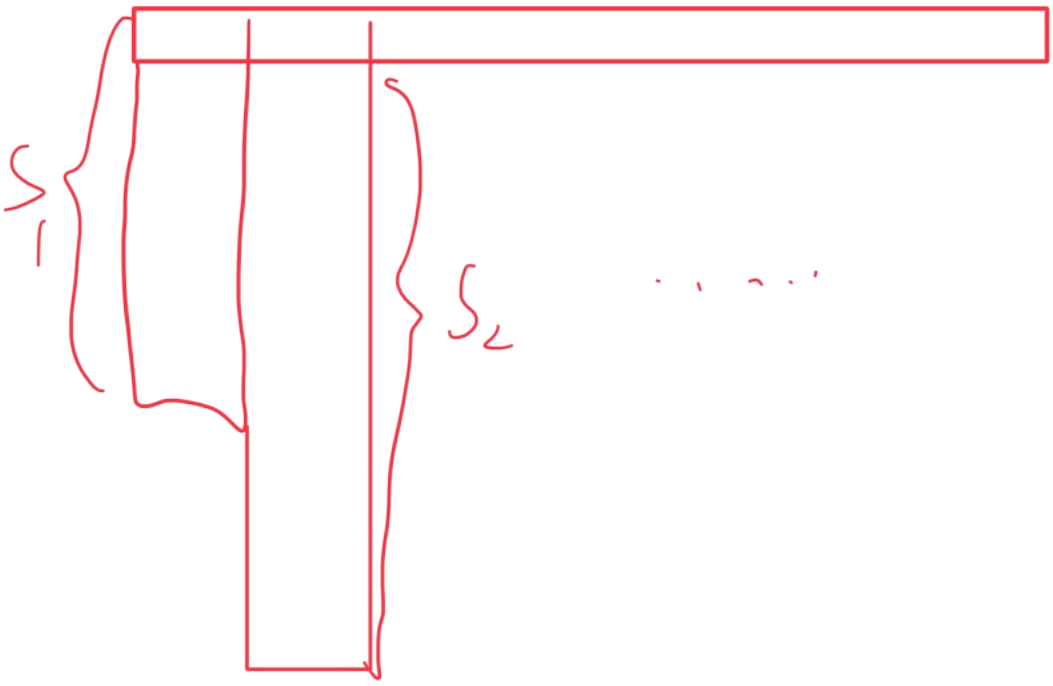
它含有多个计数 sub list length 的 variable

$h(x_1)$ takes $O(1)$

so $h(x_1)$.length takes $O(1)$

Problem 7. A multimap is a data structure that allows for multiple values to be associated with the same key. The method GET(k) should return all the values associated with key k. Describe an implementation where this method runs in $O(1+s)$ expected time, where s is the number of values associated with k.

Using Hash Function takes O(1)



Problem 8. Suppose that you have a group of n people and you would like to know if there are two people that share a birthday. Design an $O(1)$ time algorithm that given the information about the n people's birthdays, finds a pair that shares a birthday, or reports that no such pair exists.

- need 365 for Hash Table
- 只要找到是否有人有重复的 birthday 就行
- 不用找到所有的 pairs.

Problem 9. In computational linguistics, texts (such as a book or an article) are modelled as a sequence of words. A k-gram is a sequence of k consecutive words. A common task in language modelling requires that we compute the frequency of all k-grams that appear in the text.

Given a text with n words, design an $O(n)$ expected time algorithm that computes the frequency of all k-grams that appear at least once in the text.

和6差不多，都是在 Separate chaining 的基础上加一个 counter