Thursday, March 22, 2018

Part (a): $h(k) = k \% 7$

① Insert 15 ($15 \% 7 = 1$) ✓

② Insert 22 ($22 \% 7 = 1$)

$1 + 1^2 = 2$ ✓

③ Insert 36 ($36 \% 7 = 1$)

$1 + 1^2 = 2$ ×

$1 + 2^2 = 5$ ✓

④ Remove 22:

Modify 22 by "R" symbol

⑤ Find 36:

sequence:

index	1	2	5
element	15	R	36

return h[5]

⑥ Insert 10:

∵ Inserting 10 into the original hashtable would make the load factor > 1/2

∴ Resize the hashtable to 11 and rehash all the existing keys

∴ The final hashtable would look like one in the right

	①	②	③	④	⑥
0					
1	15	15	15	15	
2		22	22	~~22~~ R	
3					36
4					15
5			36	36	
6					
7					
8					
9					
10					10

$15 \% 11 = 4$

$36 \% 11 = 3$

$10 \% 11 = 10$

Part (b):

- $h_1(k) = k \% 7$
- $h_2(k) = 3 - (k \% 3)$

① Insert 15 ($15 \% 7 = 1$) ✓

② Insert 22 ($22 \% 7 = 1$)

$3 - (22 \% 3) = 2$

($\therefore 1 + 2 = 3$) ✓

③ Insert 36 ($36 \% 7 = 1$)

$3 - (36 \% 5) =$

($\therefore 1 + 3 = 4$)

④ Remove 22

Modify 22 by "R" symbol

⑤ Find 36: sequence

index	1	4
element	15	36

return h[4]

⑥ Insert 10:

($10 \% 7 = 3$)

	①	②	③	④	⑥
0					
1	15	15	15	15	15
2					
3		22	22	~~22~~ R	10
4			36	36	36
5					
6					

Part (c):

If there are 27 items to search and $\frac{2m}{3}$ items are set to true. Therefore, the number of false positives would be $27 \times (\frac{2}{3})^3 = 8$ by linearity of expectation.

Analysis:

1. Large test: Complete Works of Shakespeare
2. Random test #1
3. Random test #2

- **What test files did you use (describe them)?**
  - Large test: Complete Works of Shakespeare
  - Random test #1: random words and punctuations
  - Random test #2: random words and punctuations
- **What was the capacity of your cache for each test?**
  - Large test: 1000
  - Random test #1: 1000
  - Random test #2: 1000
- **What was the total number of rotations for each file?**
  - Large Test:
    - leftRotate(): 2545231
    - rightRotate(): 2373155
    - Total rotations: 4918386
  - Random test #1
    - leftRotate(): 7473
    - rightRotate(): 7678
    - Total rotations: 15151
  - Random test #2
    - leftRotate(): 4492
    - rightRotate(): 4477
    - Total rotations: 6624
- **What was the size of each file?**
  - Large test: 900987 words
  - Random test #1: 1216 words
  - Random test #2: 813 words
- **What was the average number of rotations per item (that is, your answer to 3, divided by your answer to 4)?**
  - Large test:
    - Average leftRotate(): 2.825
    - Average rightRotate(): 2.634

        - Average rotations: 5.459
    - Random test #1:
        - Average leftRotate(): 6.145
        - Average rightRotate(): 6.314
        - Average rotations: 12.459
    - Random test #2:
        - Average leftRotate(): 5.525
        - Average rightRotate(): 5.507
        - Average rotations: 11.032

- **How many items did you have to remove from the cache? This occurs when you bring a new item into the cache, which is referred to as a `cache miss`. Caches are designed in such a way as to minimize cache misses.**
    - Large Test: deleteMinLeaf(): 762602
    - Random test #1: deleteMinLeaf(): 195
    - Random test #2: deleteMinLeaf(): 0
- **Was there a noticeable difference between the two moderate-sized tests? Explain why there was or was not a difference.**
    - There was not a noticeable difference in two moderate-sized test, because they are both randomly generated. Therefore, the number of average rotations is very similar between two tests. Also, since there is almost no repetition of words in randomly generated tests, the cache is not working the most effectively.
- **Include any other interesting analysis your test cases revealed.**
    - If there is no repetition in the test files, cache would not work the most effectively with this type of text, because all the words just get inserted into the cache. However, if there are repetitions in the test files, cache would work very effectively because it would not insert repetitive words, so that cache would only contain unique elements.
    - For the **Random test #2,** deleteMinLeaf() is not called because the cache's capacity is larger than the size of the test file. Therefore, the condition "size > capacity" would not be called.