W08 - Hash Tables and Cuckoo Hashing

Due 3 Mar at 23:59 **Points** 20 **Questions** 3
Available 27 Feb at 8:00 - 3 Mar at 23:59 **Time limit** None
Allowed attempts Unlimited

Instructions

Content and Background

This quiz relates to the content covered in the course up till now. It may also draw upon supporting knowledge and skills expected from a CS sophomore. Please make sure that you are up to date on the coursework before attempting the quiz.

Difficulty

This quiz is equivalent to an in-class exercise. Have pen and paper ready and be prepared to work on challenging problems.

Discussion

Please use Discussion Forums to discuss any of the questions. Do not reveal your answers.

This quiz was locked 3 Mar at 23:59.

Attempt history

	Attempt	Time	Score
KEPT	Attempt 3	73 minutes	20 out of 20
LATEST	Attempt 3	73 minutes	20 out of 20
	Attempt 2	354 minutes	0 out of 20 *
	Attempt 1	67 minutes	7 out of 20 *

* Some questions not yet graded

ⓘ Correct answers are hidden.

Score for this attempt: **20** out of 20

Submitted 3 Mar at 17:18

This attempt took 73 minutes.

Question 1

5 / 5 pts

When we say that hash tables perform addition, removal, and lookup in O(1) time, in what sense do we mean O(1)?

- ○ Expected time
- ○ Amortized time
- ◉ Expected and amortized time
- ○ None of the above

"Cuckoo hashing is a scheme [...] for resolving hash collisions of values of hash functions in a table, with worst-case constant lookup time. The name derives from the behavior of some species of cuckoo, where the cuckoo chick pushes the other eggs or young out of the nest when it hatches; analogously, inserting a new key into a cuckoo hashing table may push an older key to a different location in the table." **[1]** **(https://en.wikipedia.org/wiki/Cuckoo_hashing)** You can read more about cuckoo hashing at **[2]**

(https://cs.stanford.edu/~rishig/courses/ref/l13a.pdf) , [3] (https://www.geeksforgeeks.org/cuckoo-hashing/) , and [4] (https://programming.guide/cuckoo-hashing.html) . Once done, attempt the problems below.

Image of Asian Koel, a type of cuckoo, from [5] (https://en.wikipedia.org/wiki/Asian_koel)

Question 2

5 / 5 pts

Can cuckoo hashing be generalized to use any fixed number of hash tables? What changes do you see in the implementation? Discuss.

Your answer:

Yes, Cuckoo hashing can be generalized to use any arbitrary, but fixed number of hash tables.
In this implementation, each element can have more than two possible hash functions, therefore, more possible hash values [for n hash tables there are n hash functions]. Therefore, each element can be stored in either of the multiple hash tables. For example, if there are 5 hash tables, then for one element there will be 5 hash functions, so each element can be inserted/moved into either of the 5 possible hash tables. So by using more hash tables, one can be expected to utilize more of the capacity of the hash table, therefore, increasing the spread and reducing the likelihood of rehashing for any given number of insertions, but by

sacrificing some lookup and insertion speed [they would generally be expected to take longer as well].

The implementation changes for cuckoo hashing with multiple hash tables would be somewhat similar to cuckoo hashing with two hash tables. However, more hash tables would mean more hash functions would be needed and more hash values would be computed for any element. Moreover, there should be some way to store the different hash values for a single element and should keep track of the hash values used while inserting/moving an element while keeping track of which values for any hash table have been filled. This would add up to the lookup and inserting/relocating cost, and would require more computations, and memory. Therefore, by this implementation, our complexity would increase significantly upon adding more and more hash tables.

However, there would also be higher load factors, as a single element would be stored in various hash tables. More hash tables with more hash functions would also mean there would be fewer chances of an element being shifted by another element during insert.

Question 3

10 / 10 pts

Consider a cuckoo hashing scheme using two hash tables of size 10 and using hash functions h_1 and h_2. Apply this scheme to insert the following sequence of keys to the initially empty tables: {29, 38, 51, 41, 36, 99, 123, 9, 4, 68, 149, 59}. The values of h_1 and h_2 for the keys are given below.

key,k	29	38	51	41	36	99	123	9	4	68	149	59
$h_1(k)$	9	8	1	1	6	9	3	9	4	8	9	9
$h_2(k)$	6	7	2	9	9	0	5	2	1	4	0	2

Indicate the state of the two tables after each insertion. Write -1 if it does not exist.

Index	Table 1	Table 2
0	-1	149
1	51	-1
2	-1	59
3	123	-1
4	4	-1
5	-1	-1
6	36	29
7	-1	38
8	68	-1
9	99	41

Answer 1:

-1

Answer 2:

149

Answer 3:

51

Answer 4:

-1

Answer 5:

-1

Answer 6:

59

Answer 7:

123

Answer 8:

-1

Answer 9:

4

Answer 10:

-1

Answer 11:

-1

Answer 12:

-1

Answer 13:

36

Answer 14:

29

Answer 15:

-1

Answer 16:

38

Answer 17:

68

Answer 18:

-1

Answer 19:

99

Answer 20:

41

Quiz score: **20** out of 20