CSC-361: Hashing and Hash Tables

Please work in a pair. Only one group members needs to submit; however, make sure all names are listed.

1. Indicate true / false for each statement.

   (a) It is not possible for a chaining hash table to function as one linked list.

   (b) Consider a chaining hash table T of size $2n$, but currently it only contains n items. Under the assumption that the hash function for T uniformly distributes items, then the probability of inserting the next item without a collision is 0.5.

   (c) The search operation in a cuckoo hash table may require 3 hash operations.

2. Suppose that the keys A through G with the hash values given below, are inserted in some order into an initially empty table of size 7 using a linear (sequential) probing table (with no resizing).

key	A	B	C	D	E	F	G
hash($M = 7$)	2	0	0	4	4	4	2

   Which of the following orders could not possibly result from inserting these keys?

   (a) $E\ F\ G\ A\ C\ B\ D$
   (b) $C\ E\ B\ G\ F\ D\ A$
   (c) $C\ B\ G\ A\ F\ E\ D$
   (d) $B\ D\ F\ A\ C\ E\ G$
   (e) $C\ B\ A\ G\ D\ F\ E$
   (f) $C\ G\ B\ A\ D\ E\ F$
   (g) $F\ G\ B\ D\ A\ C\ E$
   (h) $G\ E\ C\ A\ D\ B\ F$

3. Let H be a hashtable of integers where collision is resolved using dual-array Cuckoo hashing as shown below. Assume H defines the first hash function `hash1(key) = key % 9` as well as second hash function `hash2(key) = key % 5`. Show the resulting hashtable after inserting the values {`15 7 10 0 6 1`} in sequence. (6 points)

   For the result of inserting into the hashtable. (Will be graded.)

`hash1(key) = key % 9`

index	H[index]
0	
1	
2	
3	
4	
5	
6	
7	
8	

`hash2(key) = key % 5`

index	H[index]
0	
1	
2	
3	
4	

4. Consider a hashtable of length m with linear probing used to resolve collisions. What is the *worst* case number of collisions that may occur adding m elements to the table? Explain this worst case scenario and show all work required to derive your expression.

5. Explain why we cannot use binary search to `search` for elements in a chain in a chaining hash table.