COMP.4040 – Quiz 6 *Su2019*

Name: (Print) ______________________________
Due: June 26, 2019

1. (5 points) Given the following input (3412, 3413, 1741, 3269, 2909, 6291, 6373, 5129) and the hash function h(k) = k mod 10, which of the following statement(s) are true? Choose all correct ones.

A. 3269, 2909, 5129 hash to the same value

B. 3412 and 3413 hash to the same value

C. 1741, 6291 hash to the same value

D. 3413, 3269, 6291, 6373 each hashes to a different value

2. (5 points) The keys 14, 18, 33, 4, 3, 23, 25 and 5 are inserted into an initially empty hash table in this given order. The hash table has 10 slots and uses chaining with hash function h(k) = k mod 10. What is the hash table after inserting all keys? (multiple numbers in the same slot represents a linked list to chain the numbers together in that order)

0	
1	
2	
3	3, 23, 33
4	4, 14
5	5, 25
6	
7	
8	18
9	

A

0	
1	
2	
3	23, 3, 33
4	4, 14
5	5, 25
6	
7	
8	18
9	

B

0	
1	
2	
3	33, 23, 3
4	14, 4
5	25, 5
6	
7	
8	18
9	

C

0	
1	
2	
3	33, 3, 23
4	14, 4
5	25, 5
6	
7	
8	18
9	

D

3. (5 points) The keys 14, 18, 33, 4, 3, 23, 25 and 5 are inserted into an initially empty hash table in this given order. The hash table has 10 slots and uses open addressing with hash function h(k) = k mod 10 and linear probing. What is the hash table after inserting all keys?

0	
1	
2	
3	23
4	4
5	5
6	
7	
8	18
9	

A

0	
1	
2	5
3	33
4	14
5	4
6	3
7	23
8	18
9	25

B

0	
1	
2	
3	33
4	14
5	25
6	
7	
8	18
9	

C

0	5
1	
2	
3	33
4	14
5	4
6	3
7	23
8	18
9	25

D

4. (5 points) (1) What is the load factor in Problem 2 above?

(2) What is the load factor in Problem 3 above?

4. (20 points) **Design and Analysis of an Algorithm**

Consider an unsorted array A of n integers; design an efficient algorithm that accepts A, n and s as the inputs and determines if the array contains two integers such that they add up to a specific target number s. That is: if we can find $A[i] + A[j] == s$ ($1 \leq i, j \leq n$, $i \neq j$), the algorithm should return TRUE, otherwise return FALSE.

Design requirement:

- the *efficient* algorithm you are going to design should provide an **O(nlgn)** running time, rather than an $O(n^2)$ running-time solution.
- To keep your answers brief, you may use any algorithms that we have learned from lectures and the textbook as subroutines (this means you do NOT need to re-write those algorithms, just call them with the proper input/output).

(1) (12 points) Algorithm Pseudocode *(please use textbook conventions)*:

(2) (8 points) What is the running time of the algorithm that you designed? Justify your answer.