Name: (Print) PHONG VO
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 output = 9

B. 3412 and 3413 hash to the same value

C. 1741, 6291 hash to the same value output = 1

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)

A.

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

B.

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

C.

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

D.

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

$h(14) = h(4) = 4$

$h(18) = 8$

$h(33) = h(3) = h(23) = 3$

$h(25) = h(5) = 5$

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?

$h(23,4) = (23+4) \bmod 10 = 7$

$h(25,0) = (25+0) \bmod 10 = 5$ (collision)

$h(25,1) = (25+1) \bmod 10 = 6$ (collision)

$h(25,2) = (25+2) \bmod 10 = 7$ (collision)

$h(25,3) = (25+3) \bmod 10 = 8$ (collision)

$h(25,4) = (25+4) \bmod 10 = 9$

A

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

B

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

C

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

D

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

$h(14,0) = (14+0) \bmod 10 = 4$

$h(18,0) = (18+0) \bmod 10 = 8$

$h(33,0) = (33+0) \bmod 10 = 3$

$h(4,0) = (4+0) \bmod 10 = 4$ (collision)

$h(4,1) = (4+1) \bmod 10 = 5$

$h(3,0) = (3+0) \bmod 10 = 3$ (collision)

$h(3,1) = (3+1) \bmod 10 = 4$ (collision)

$h(3,2) = (3+2) \bmod 10 = 5$ (collision)

$h(3,3) = (3+3) \bmod 10 = 6$

$h(23,0) = 23 \bmod 10 = 3$ (collision)

$h(23,1) = (23+1) \bmod 10 = 4$ (collision)

$h(23,2) = (23+2) \bmod 10 = 5$ (collision)

$h(23,3) = (23+3) \bmod 10 = 6$ (collision)

$h(5,0) = (5+0) \bmod 10 = 5$ (collision)

$h(5,1) = (5+1) \bmod 10 = 6$ (collision)

$h(5,2) = (5+2) \bmod 10 = 7$ (collision)

$h(5,3) = (5+3) \bmod 10 = 8$ (collision)

$h(5,4) = (5+4) \bmod 10 = 9$ (collision)

$h(5,5) = (5+5) \bmod 10 = 0$

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

$$\alpha = \frac{n}{m} = \frac{8}{10} = .8$$

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

$$\alpha = \frac{n}{m} = \frac{8}{10} = .8$$

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

Consider an unsorted array *A* of *n* integers; design an efficient algorithm that accepts <u>*A, n* and *s*</u> 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 ($1 \le i, j \le n, i \neq j$), the algorithm should return TRUE, otherwise return FALSE.

mergesort
heap sort

<u>Design requirement</u>:

- 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)*:

```
Sum_seeking (A, n, s)
    // sort first
    MERGE_SORT (A, 1, n)          } T(n) = O(nlgn)          -3

for (i = 1 to n)
    for (j = i to n)
        if (A[i] + A[j] == s)      } T(n) = O(n)
            return TRUE                this is Θ(n²)
                                       not O(n)
return FALSE
```

Seeking (S − A[i]) vs. every element. If == , → return TRUE
if != , → return FALSE

i →

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

The total running time

$$T(n) = O(n) + O(n \lg n)$$

$$\Rightarrow T(n) = O(n \lg n)$$

−5

32/40