CS 101 Fall 2021 - Quiz 2-A

10/11/2021 - 25 Minutes **Name:** **ID number:**

Remember that your work is graded on the quality of your writing and explanation as well as the validity.

Problem 1 Notes of discussion (1 pts):

I promise that I will complete this QUIZ independently, and will not use any electronic products or paper-based materials during the QUIZ, nor will I communicate with other students during this QUIZ.

True or False: I have read the notes and understood them.

1
T

Problem 2 Single Choice(3×2pts):

The following questions are single choice questions, each question has **only one** correct answer. Select the correct answer.

Note: You should write those answers in the box below.

Question 2.1	Question 2.2	Question 2.3
c	c	a

2.1: A hash table of length 10 uses open addressing with hash function $h(k) = k \ mod \ 10$, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below:

Index	0	1	2	3	4	5	6	7	8	9
Keys			42	23	34	52	46	33		

TABLE 1. Hash table of question 1.1&1.2

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

(a) 46, 42, 34, 52, 23, 33

(b) 34, 42, 23, 52, 33, 46

(c) 46, 34, 42, 23, 52, 33

(d) 42, 46, 33, 23, 34, 52

2.2: How many different insertion sequences of the key values using the hash function $h(k) = k \ mod \ 10$ and linear probing will result in the hash table shown in Table 1?

(a) 10

(b) 20

(c) 30

(d) 40

2.3: Consider a hash table with 100 slots. Collisions are resolved using linked list. Assuming simple uniform hashing, which means each key will be hashed to each slot with the same probability, what is the probability that the first 3 slots are unfilled after the first 3 insertions?

(a) $(97 \times 97 \times 97)/100^3$

(b) $(99 \times 98 \times 97)/100^3$

(c) $(97 \times 96 \times 95)/100^3$

(d) $(97 \times 96 \times 95)/(3 \times 100^3)$

CS 101 Fall 2021 - Quiz 2-A

10/11/2021 - 25 Minutes **Name:** **ID number:**

Problem 3 Hash collisions(3+3 pts):

Suppose we want to use a hash table of size 10 to store 7 students' scores. We are using the hash function $h(k) = k \ mod \ 10$ to insert each "Student k" and are using linear probing to resolve collisions. After inserting these students' scores into the hash table, it looks like below:

Index	0	1	2	3	4	5	6	7	8	9
Keys	24				14	35	54	55	98	17

TABLE 2. Hash table of problem 3

(1) Accidentally, we spills some coffee on the hash table, rendering it illegible. To cover its incompleteness, we want to recreate the list of students' scores in the order they were given, which is the same as the order they were inserted; however, we only remember two facts about this order:

(a) 98 was the first student score to be inserted.

(b) 35 was inserted before 14.

Help us by figuring out what the order must have been. Write the order directly.

98, 35, 14, 54, 55, 17, 24

(2) We want to use a new hash function to insert all students into a new empty hash table:

$$h(k) = ((k + 7)/15 + k) \ mod \ 10$$

Collisions are resolved by using quadratic probing, with the probe function:

$$(i^2 + i)/2$$

Fill in the final contents of the hash table after the key values have been inserted in the order which is the same as question (1). We need to specify that the a/b operation means $a/b = \lfloor \frac{a}{b} \rfloor$.

Index	0	1	2	3	4	5	6	7	8	9
Keys		17	24			98	14	35	54	55

CS 101 Fall 2021 - Quiz 2-A

10/11/2021 - 25 Minutes **Name:** **ID number:**

Problem 4 True or False(3×2pts):

The following questions are True or False questions, you should judge whether each statement is true or false and write the answer(T/F) in the box below.

Question 4.1	Question 4.2	Question 4.3
T	F	T

4.1: Given an array of n elements, where each element is at most k away from its target position, insertion sort can give a $\mathbf{O}(kn)$ performance.

4.2: On average, Insertion sort performs better on large problems rather than small problems.

4.3: Whether with or without a flag, the worst case time complexity for bubble sort is always $\mathbf{O}(n^2)$

Problem 5 Sorting Implementation(2+4 pts):

The following is the implementation of a sorting algorithm.

Note: The array has its first item indexed as 1, and 'for a to b' means iterating every x where $x \in [a, b]$

```
Procedure Sort(A):
  for j = 2 to A.length:
      key = A[j]
      i = j - 1
      while i > 0 and A[i] > key:
          A[i+1] = A[i]
          i = i - 1
      A[i+1] = key
      // Mark
```

(1) Which sorting algorithm does it describe?

(2) Give a list as [1,4,2,8,5,7], we use the above procedure to sort it. Write down what will the list be like each time when the procedure meets the 'Mark'.

(1) Insertion sort.

(2) [1,4,2,8,5,7]

$[1, 2, 4, 8, 5, 7]$

$[1, 2, 4, 8, 5, 7]$

$[1, 2, 4, 5, 8, 7]$

$[1, 2, 4, 5, 7, 8]$