CS 101 Fall 2024 - Quiz 2
October, 14, 2024 - 20 Minutes **Name:** **Student ID:**

---

1. (2 points) Honor Code

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.

I will not violate the Honor Code during this quiz.

√ True ◯ False

2. (10 points) True or False

Determine whether the following statements are true or false.

(a) (1') The average-case running time for searching in a hash table is $\Theta(n)$. ◯ True √ False

Solution: It depends on λ, but it can be considered $\Theta(1)$ if we regard λ as a constant.

(b) (1') Given a hash function $h(\cdot)$, it satisfies: $h(x) = h(y) \Rightarrow x = y$ ◯ True √ False

Solution: Different x, y may map to the same hash value.

(c) (1') The worst-case running time for insertion in a hash table is $O(1)$. ◯ True √ False

Solution: $O(n)$ due to potential collisions that require chaining or open addressing to resolve.

(d) (1') Linear probing is equivalent to double hashing with a secondary hash function of $h_2(k) = 1$.

√ True ◯ False

(e) (1') Quadratic probing is equivalent to double hashing with a secondary hash function of $h_2(k) = k^2$. ◯ True √ False

(f) (1') Given an array of n elements, where each element is at most k away from its target position, insertion sort can give a $O(kn)$ performance.

√ True ◯ False

(g) (1') Insertion sort is generally more efficient for small datasets compared to large datasets.

√ True ◯ False

(h) (1') In the worst case, bubble sort requires $(n-1)$ complete passes through the array.

√ True ◯ False

(i) (1') The space complexity of bubble sort, insertion sort, and merge sort is all $O(1)$.

◯ True √ False

Solution: Merge sort has a space complexity of $O(n)$ due to the need for additional space to hold the merged subarrays.

(j) (1') In the implementation of merge sort, the counting of inversions can be achieved using a simple counter during the merging of two subarrays.

√ True ◯ False

3. (10 points) Hash table insertion simulation

Given a hash table with $M = 11$ slots and hash function $h_1(x) = (3x + 6) \bmod 11$. We want to insert integer keys $A = [33, 20, 2, 16, 23, 48, 35, 6, 31, 44]$.

(a) (5') **Linear probing hash table**

i. Suppose that collisions are resolved through linear probing. The integer key values listed below will be inserted in the order given. Write down the index of the home slot (the slot to which the key hashes before any probing) and the probe sequence (do not write the home slot again) for each key. If there's no probing, leave the cell blank.

Key Value	33	20	2	16	23	48	35	6	31	44
Home Slot	6	0	1	10	9	7	1	2	0	6
Probe Seq.							2	3	1,2,3,4	7,8

ii. Write down the content of the hash table after all the insertions.

Index	0	1	2	3	4	5	6	7	8	9	10
Keys	20	2	35	6	31		33	48	44	23	16

(b) (5') **Double hashing hash table**

i. Suppose that collisions are resolved through double hashing. The probing function is described as

$$H_i(k) = (h_1(k) + i \cdot h_2(k)) \bmod 11$$

for any give key value k in the i-th probing (i starts from 0). $h_2(k)$ is the second hash function defined as

$$h_2(k) = 7 - (k \bmod 7)$$

Write down the index of the probe sequence for each key.

Key Value	33	20	2	16	23	48	35	6	31	44
Probe Seq.							8		4	0,5

ii. Write down the content of the hash table after all the insertions.

4. (6 points) Sorting Implementation

The following is the implementation of a sorting algorithm.

```
Procedure Sort(A):
    for i=1 to A.length-1:
        for j=A.length-1 downto i:
            if A[j] < A[j-1]:
                key=A[j]
                A[j]=A[j-1]
                A[j-1]=key
        //Mark
```

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

(a) (2') Which sorting algorithm does it describe?

Index	0	1	2	3	4	5	6	7	8	9	10
Keys	20	2	6		31	44	33	48	35	23	16

Solution: Bubble sort.

(b) (4') Give a list as [1,3,2,8,5,7], we use the above procedure to sort it. Write down what will the list be like each time when the procedure meets the 'Mark'.

Solution: 1; 2; 3; 5; 8; 7
1; 2; 3; 5; 7; 8
1; 2; 3; 5; 7; 8
1; 2; 3; 5; 7; 8
1; 2; 3; 5; 7; 8