DSC 40B - Quiz 05

Name: Student ID:

About the quizzes:

- Quizzes in DSC 40B are *optional* and graded pass/fail.
- A score of 70% or higher earns a "pass" and 1.5 credits toward your final grade.
- If you don't pass, no credits are earned, but it doesn't hurt your grade.
- You have 30 minutes to complete the quiz.
- Unfortunately, we can't answer clarifying questions during the quiz. If you think a question has a bug or is unclear, please let us know in a private post on Campuswire after the quiz, and we'll take it into account when grading.

Problem 1.

Suppose a hash table has 50 bins, and 400 elements are inserted into it using a hash function that places each element into a bin uniformly at random. What is the expected number of elements in the first bin?

◯ 1 ◯ 4 ◯ 8 ◯ 50 ◯ 400

Problem 2.

Suppose a hash table is constructed so that collisions are resolved through chaining. The hash table has 10 bins, and 100 items are inserted into the hash table. True or False: it is possible for one of the bins to contain all 100 items.

◯ True

◯ False

Problem 3.

Assume that a hash table is implemented using chaining to resolve collisions. The hash table stores n numbers. Suppose a membership query is made for 42. True or False: more than one bin in the hash table may be checked with a linear search during the query.

◯ True

◯ False

Problem 4.

Suppose a hash table stores n elements and resolves collisions by chaining. What is the **worst case** time complexity of a membership query? You may assume that computing the hash of an element takes $\Theta(1)$ time.

◯ $\Theta(1)$ ◯ $\Theta(\log n)$ ◯ $\Theta(n)$ ◯ $\Theta(n \log n)$

Problem 5.

Suppose we are building a hash table with 1000 bins, numbered 0 through 999, to store nonnegative integers. Each time we insert a number, we place it in the bin whose number is the **last digit** of the number. For example, 1234 goes into bin 4, and 57 goes into bin 7.

True or False: this is a good hash function for this hash table.

◯ True

◯ False

Problem 6.

Suppose you insert 11 elements into a hash table with 20 bins. True or False: there must be **exactly** 9 empty bins, no more, no less, since the number of elements being hashed is less than the number of bins in the hash table by nine.

◯ True

◯ False

Problem 7.

Suppose `arr` is a list of n integers. What is the **expected** time complexity of the code below? Remember that Python's `set` is implemented using a hash table.

```
def find_duplicates(arr):
    seen = set()
    duplicates = set()
    for num in arr:
        if num in seen:
            duplicates.add(num)
        else:
            seen.add(num)
    return duplicates
```

◯ $\Theta(1)$ ◯ $\Theta(n)$ ◯ $\Theta(n \log n)$ ◯ $\Theta(n^2)$

Problem 8.

Consider the function below, where `A` and `B` each contain n numbers.

```
def intersection(A, B):
    common = set()
    for x in A:
        if x in B:
            common.add(x)
    return common
```

a) What is the **expected** time complexity of `intersection` if `B` is a Python `set`?

◯ $\Theta(1)$ ◯ $\Theta(n)$ ◯ $\Theta(n \log n)$ ◯ $\Theta(n^2)$

b) What is the **expected** time complexity of `intersection` if `B` is a Python `list`? Remember that `x in B` does a linear search when `B` is a list.

◯ $\Theta(1)$ ◯ $\Theta(n)$ ◯ $\Theta(n \log n)$ ◯ $\Theta(n^2)$

Problem 9.

Suppose a hash table stores n elements, and inserting one more element causes it to grow: it allocates a larger table and copies (rehashes) every element into the new table. What is the time complexity of this one insertion?

◯ $\Theta(1)$ ◯ $\Theta(\log n)$ ◯ $\Theta(n)$ ◯ $\Theta(n^2)$

Problem 10.

For each situation below, which data structure has the better time complexity? Assume that the hash table uses a good hash function and that the binary search tree is balanced.

a) You will insert n numbers, and make n membership queries.

◯ Hash table ◯ Balanced binary search tree

b) You will insert n numbers. After each insertion, you will be given a number x (a different one each time), and must find the smallest number inserted so far that is larger than x.

◯ Hash table ◯ Balanced binary search tree