CS 101 Fall 2023 - Quiz 2

10/23/2023 - 20 Minutes **Name:** **ID number:**

---

1. (1 points) Notes of discussion

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 have read and understood the notes.

√ True ○ False

2. (5 points) True or False

Determine whether the following statements are true or false.

(a) (1') If $f(n) = n^{\log n}$ then for all $\alpha \geq 1$, we have $f(n) = \omega(n^\alpha)$.

√ True ○ False

Solution: $\lim_{n\to\infty} \frac{n^{\log n}}{n^\alpha} = \lim_{n\to\infty} n^{\log n - \alpha} = +\infty$

(b) (1') $f(n) = o(g(n)) \implies f(n) + g(n) = \Theta(g(n))$.

√ True ○ False

Solution: $\lim_{n\to\infty} \frac{f(n)+g(n)}{g(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} + \lim_{n\to\infty} \frac{g(n)}{g(n)} = 0 + 1 = 1$

(c) (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.

(d) (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.

(e) (1') If we want to search for an element by value and delete this element, hash table with a good hash function will perform better than array and linked list.

√ True ○ False

Solution: Searching plus deletion is $O(n)$ for array and linked list, but $\Theta(1)$ for hash table.

3. (4 points) Analysing the Time Complexity of a C++ Function

NOTE: Please clearly demonstrate your complexity analysis. The answer of the total complexity alone only accounts for 1pt.

```
std::vector<double> foo(std::vector<double> a) {
    int n = a.size();
    double max = a[0];
    for (int i = 1; i < n; ++i) {
        if (max < a[i]) {
            max = a[i];
            std::reverse(a.begin(), a.begin() + i + 1); // Theta(i)
        }
    }
    return a;
}
```

What is the average-case time complexity of of `foo` (in the form of $\Theta(f(n))$, where n is the size of the `vector a`)? You may assume that all operations of one or two single `int` or `double` variables take *constant time*.

Hint: What is the rough probability that a_i is the largest among $a_1, a_2, ..., a_i$ if all elements in `vector a` is independent and evenly distributed? You can recall the lecture sildes and don't need to prove it.

Solution: The average-case time complexity is $\Theta(n)$. (1pt)

The probability of entering the `if` body in the i-th loop is $\frac{1}{i}$, because it is the probability that a_i is the largest one among $a_1 \sim a_i$. (1pt)

The time complexity of the `if` body in the i-th loop is $\Theta(i)$, because `max = a[i]` is $\Theta(1)$ and `std::reverse(a.begin(), a.begin() + i + 1)` is $\Theta(i)$. (1pt)

The average-case time complexity of the i-th loop is $\frac{1}{i}\Theta(i) + \frac{i-1}{i}\Theta(1) = \Theta(1)$, and there are $\Theta(n)$ loops, so the total complexity is $\Theta(n)$. (1pt)

4. (6 points) Quadratic Probing Hash Table

Given a hash table with $M = 8$ slots and hash function $h(k) = (\lfloor \frac{k+7}{15} \rfloor + k) \bmod 8$. We want to insert integer keys $A = [17, 24, 98, 14, 33, 25, 58]$.

(a) (3') Suppose that collisions are resolved through quadratic probing with the probe function:

$$\frac{i^2 + i}{2}$$

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	17	24	98	14	33	25	58
Home Slot	2	2	1	7	3	3	6
Probe Sequence		3			4	4, 6	7,1,4,0

(b) (3') Write down the content of the hash table after all the insertions.

Index	0	1	2	3	4	5	6	7
Keys	58	98	17	24	33		25	14