CS 101 Fall 2022 - Quiz 2-B

9//2022 - 25 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.

(a) (1') True or False: I have read and understood the notes.

problem 1

2. (8 points) True or False

Determine whether the following statements are true or false.

problem (a)	problem (b)	problem (c)	problem (d)	problem (e)	problem (f)	problem (g)	problem (h)
F	T	T	F	F	T	F	F

(a) (1') $f(n) = O(g(n)) \land f(n) = O(h(n)) \implies f(n) = O(h(n)g(n))$.

(b) (1') If $f(n) = \log_2 n$ then for all $0 < \alpha \leq 1$, we have $f(n) = O(n^\alpha)$.

(c) (1') For an algorithm **A**, it is *possible* that the worst-case running time is $O(n)$ and the best-case running time is $\Omega(n)$.

(d) (1') For an algorithm **B**, it is *impossible* that the worst-case running time is $\Omega(n)$ while the best-case running time is $O(\sqrt{n})$.

(e) (1') The number of collisions in a hash table solely(only) depends on the table capacity and the hash function.

(f) (1') The worst-case running time for insertion in a hash table is $O(n)$.

(g) (1') Hash tables using open addressing are better implemented with linked lists than with arrays because key values can be added or deleted quickly.

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

3. (8 points) Hash Table Insertions and Deletions

Consider a empty hashtable of capacity 7 and with hash function $h(k) = (2k + 5) \mod 7$. Collisions are resolved by quadratic probing with the probing function $H_i(k) = (h(k) + i^2) \mod 7$, paired with lazy erasing. We will give 8 instructions (`Insert/Delete/Search key_value`). For `Insert/Delete` instructions, you need to fill the hash table after each instruction. For `Search` instructions, write down probing sequence(index). Use '`D`' to indicate that the bin has been marked as deleted.

(a) (1') `Insert 13`

Index	0	1	2	3	4	5	6
Key Value				13			

(b) (1') `Insert 27`

Index	0	1	2	3	4	5	6
Key Value				13	27		

(c) (1') `Insert 34`

Index	0	1	2	3	4	5	6
Key Value	34			13	27		

(d) (1') `Delete 27`

Index	0	1	2	3	4	5	6
Key Value	34			13	D		

(e) (1') `Insert 8`

Index	0	1	2	3	4	5	6
Key Value	34	8		13	D		

(f) (1') `Delete 13`

Index	0	1	2	3	4	5	6
Key Value	34	8		D	D		

(g) (1') `Insert 24`

Index	0	1	2	3	4	5	6
Key Value	34	8		D	24		

(h) (1') `Search 34`

Solution: 3, 4, 0

4. (4 points) Balancing the Running Time

Suppose that the running time of an algorithm is $T(n) = B^2 + \frac{n^2}{B^2}$, where n is the size of the input and B is a unknown fixed parameter that might not be independent of n.

(a) (2') What is the asymptotic tight upper bound of $T(n)$ if $B = n$

(b) (2') Try to find a function $g(n)$ such $B = g(n)$ minimizes the order of growth of $T(n)$.

Solution:

part 1

$$T(n) = n^2 + \frac{n^2}{n^2} = n^2 + 1 = \Theta(n^2)$$

$O(n^2)$ **is also okay**

part 2

$$T(n) \geq 2\sqrt{B^2 \times n^2/B^2} = 2n$$

LHS equals to RHS iff $B^2 = \frac{n^2}{B^2}$, thus $B = \sqrt{n}$ and $T(n) = 2n = \Theta(n)$.

$O(n)$ **is also okay**