CSE3358 Test 1
10/05/04
12:30 PM - 1:50 PM

Name:————————————————-

- This test is closed book, closed notes, open minds...
- There are 5 Problems. For problems with several parts, each part is on a separate page.
- Scratch pages are provided at the end. Scratch pages **will not** be graded.
- Make sure your answer is clear, especially when you provide a pseudocode or describe an algorithm.
- Do not leave unanswered questions even if you think you do not have the complete answer. Partial credit might be given.
- Read all questions first. This will help you identify which questions you can easily answer first.

Problem 1: 10 points
Problem 2: 10 points
Problem 3: 12 points
Problem 4: 8 points
Problem 5: 20 points

————————————-

Total: 60 points

Problem 1: Recurrences (10 points, 2 points each)
For each of the following divide-and-conquer algorithms, specify the letter corresponding to the recurrence governing its running time and the number corresponding the its running time.

Algorithm	Recurrence	Solution
Merge sort (worst case)		
Binary Search (worst case)		
Randomized Quicksort (expected)		
Strassen's matrix multiplication (worst case)		
Quicksort (worst case)		

letter	Recurrence
a	$T(n) = 7T(n/2) + \Theta(n^2)$
b	$T(n) = T(n/2) + \Theta(1)$
c	$T(n) = 2T(n/2) + \Theta(n^2)$
d	$T(n) = 2T(n/2) + \Theta(n)$
e	$T(n) = T(n/b) + T((1-1/b)n) + \Theta(n)$ $b > 2$
f	$T(n) = 2T(n/2) + \Theta(1)$
g	$T(n) = T(n-1) + \Theta(n)$

number	solution
1	$\Theta(n)$
2	$\Theta(\log n)$
3	$\Theta(n^{\log_2 7})$
4	$\Theta(1)$
5	$\Theta(n^2)$
6	$\Theta(n \log n)$

Problem 2: TRUE / FALSE questions (10 points, 2 points each)
For each of the following statements, indicate whether it is true or false **and explain why**.

T F There exists a constant $n_0 \geq 1$, such that, for every $n \geq n_0$, there is an array of n elements on which insertion sort runs faster than merge sort.

T F Consider the BUILD-HEAP algorithm we saw in class:

BUILD-HEAP
for i←n downto 1
    do HEAPIFY(A,i)

On an array of n elements this code runs in $\Theta(n \log n)$ in the worst case because there are $\Theta(n)$ calls to HEAPIFY, and the worst case time for any call to HEAPIFY is $\Theta(\log n)$.

T F The average time for a search operation in a hash table with chaining is linear in the number of elements in the hash table.

T F There exists a comparison sort algorithm that can sort 6 numbers by making at most 9 comparisons.

T F An adversary can force quicksort to run in $\Omega(n^2)$ by providing an array of size n that is already sorted or reverse-sorted.

Problem 3: Choice of data structures
You are in space and many UFOs are coming your way. Although UFOs are "unidentified", you can identify them by their transmitted keys. Every time you intercept a UFO signal, you can insert its key into some data structure together with some data regarding the UFO. For each of the following situations, indicate your choice of a data structure that would best perform the job and explain why.

(a) (3 points) You send the information to Earth. A group of scientists on Earth are interested in retrieving information on the most recently detected UFOs first.

(b) (3 points) You send the information to Earth. A group of scientists on Earth are interested in compiling a history of UFOs by observing their keys one at a time in the order UFOs where sent.

(c) (3 points) You send the information to Earth. A group of scientists on Earth are suspecting that the key of a UFO represents the date the UFO was manufactured. They are interested in retrieving information of the most technologically advanced UFOs first.

(d) (3 points) You discovered that in an attempt to confuse the human race, a UFO sets its key at random. However, you also discovered that these keys are uniformly distributed over some range. You send the information to Earth. A group of scientists on Earth are interested in performing a lot of operations most of them consist of checking whether a particular key has been observed or not.

Problem 4: Hash functions and Heaps (8 points)
Give an appropriate hash function for the following two situations in (a) and (b):

(a) (2 points) The keys are names of persons in the same family but interpreted as numbers in radix 128. The family name is more than p characters. The size of the table is $m = 128^p$.

(b) (2 points) Nothing is known about the keys in advance, but we require the computation of the hash function to be very fast, and $m = 501$.

(c) (2 points) Show the heap after BUILD-HEAP(A) is performed on $A = [4, 1, 3, 2, 16, 9, 10, 14, 8, 7]$.

(d) (2 points) Show the heap of part (c) after performing an EXTRACT-MAX(A) operation.

Problem 5 (20 points)

(a) (5 points) Let A be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A. Therefore, a sorted array has 0 inversions. List the 5 inversions in the array $A = [2, 3, 8, 6, 1]$.

(b) (5 points) In your opinion, what is the relation between the number of inversions in an array and the running time of insertion sort on that array? Let n be the length of the array, I be the number of inversions, what is the running time $T(n)$ of insertion sort in terms of n and I? Use Θ notation. Explain your answer. *Hint: Why don't you try insertion sort on the array above?*

(c) (5 points) Design an algorithm to print all inversions. Provide (i) Pseudocode and (ii) running time analysis. You will not be penalized if your algorithm is not efficient as long as the pseudocode and its corresponding time analysis are correct.

(d) (5 points) Show that any algorithm for part (c) has a worst-case running time of $\Omega(n^2)$.