Practice Quiz 1

- Do not open this quiz booklet until you are directed to do so. Read all the instructions first.
- When the quiz begins, write your name on every page of this quiz booklet.
- The quiz contains five multi-part problems. You have 80 minutes to earn 80 points.
- This quiz booklet contains **13** pages, including this one. An extra sheet of scratch paper is attached. Please detach it before turning in your quiz.
- This quiz is closed book. You may use one handwritten A4 or $8\frac{1}{2}'' \times 11''$ crib sheet. No calculators or programmable devices are permitted.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Do not put part of the answer to one problem on the back of the sheet for another problem, since the pages may be separated for grading.
- Do not waste time and paper rederiving facts that we have studied. It is sufficient to cite known results.
- Do not spend too much time on any one problem. Read them all through first, and attack them in the order that allows you to make the most progress.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Good luck!

Problem	Points	Grade	Initials
1	11		
2	19		
3	10		
4	20		
5	20		
Total	80		

Name: ______________________________

Circle your recitation letter and the name of your recitation instructor:

Brian	11	12	2	Jen	12	1

Problem 1. Recurrences [11 points]

Solve the following recurrences. Give tight, i.e. $\Theta(\cdot)$, bounds.

(a) $T(n) = 81T(\frac{n}{9}) + n^4 \lg n$ **[2 points]**

(b) $T(n) = 2T(\frac{n}{2}) + \frac{n}{\lg n}$ **[2 points]**

(c) $T(n) = 4T(n/3) + n^{\log_3 4}$ **[2 points]**

(d) Write down and solve the recurrence for the running time of the DETERMINISTIC SELECT algorithm using groups of 3 (rather than 5) elements. The code is provided on the last page of the exam. **[5 points]**

Problem 2. True or False, and Justify [19 points]

Circle **T** or **F** for each of the following statements, and briefly explain why. The better your argument, the higher your grade, but be brief. Your justification is worth more points than your true-or-false designation.

(a) T F Given a set of n integers, the number of comparisons necessary to find the maximum element is $n - 1$ and the number of comparisons necessary to find the minimum element is $n - 1$. Therefore the number of comparisons necessary to simultaneously find the smallest and largest elements is $2n - 2$. **[3 points]**

(b) T F There are n people born between the year 0 A.D. and the year 2003 A.D. It is possible to sort all of them by birthdate in $o(n \lg n)$ time.

(c) **T F** There is some input for which RANDOMIZED QUICKSORT always runs in $\Theta(n^2)$ time. **[3 points]**

(d) **T F** The following array A is a Min Heap:

$$2 \quad 5 \quad 9 \quad 8 \quad 10 \quad 13 \quad 12 \quad 22 \quad 50$$

[3 points]

(e) **T F** If $f(n) = \Omega(g(n))$ and $g(n) = O(f(n))$ then $f(n) = \Theta(g(n))$. **[3 points]**

(f) **T F** ~~Let k, i, j be integers, where $k > 3$ and $1 \leq i, j \leq k$. Let h^k_{ij} be the hash function mapping a k-bit integer $b_1 b_2 \ldots b_k$ to the 2-bit value $b_i b_j$. For example, $h^8_{31}(\mathbf{0}0\mathbf{1}01011) = 10$. The set $\{h^k_{ij} : 1 \leq i, j \leq k\}$ is a universal family of hash functions from k-bit integers into $\{00, 01, 10, 11\}$~~. **[4 points]**

We have not covered Hashing yet this semester. Expect a problem of equivalent length and difficultly.

Problem 3. Short Answer [10 points]

Give brief, but complete, answers to the following questions.

(a) Explain the differences between average-case running time analysis and expected running time analysis. For each type of running time analysis, name an algorithm we studied to which that analysis was applied.

[5 points]

(b) ~~Suppose we have a hash table with $2n$ slots with collisions resolved by chaining, and suppose that $n/8$ keys are inserted into the table. Assume each key is equally likely to be hashed into each slot (**simple uniform hashing**). What is the expected number of keys for each slot? Justify your answer.~~ **[5 points]**

We have not covered Hashing yet this semester. Expect a problem of equivalent length and difficultly.

Problem 4. Checking Properties of Sets [20 points]

In this problem, more efficient algorithms will be given more credit. Let S be a finite set of n positive integers, $S \subset \mathbb{Z}^+$. You may assume that all basic arithmetic operations, i.e. addition, multiplication, and comparisons, can be done in unit time. In this problem, partial credit will be given for correct but inefficient algorithms.

(a) Design an $O(n \lg n)$ algorithm to verify that:

$$\forall\, T \subseteq S, \quad \sum_{t \in T} t \geq |T|^3.$$

In other words, if there is some subset $T \subseteq S$ such that the sum of the elements in T is less than $|T|^3$, the your algorithm should output "no". Otherwise, it should output "yes". Argue (informally) that your algorithm is correct and analyze its running time. **[10 points]**

(b) In addition to S and n, you are given an integer $k, 1 \le k \le n,$. Design a more efficient (than in part **(a)**) algorithm to verify that:

$$\forall\, T \subseteq S,\ |T| = k, \quad \sum_{t \in T} t \ge k^3.$$

In other words, if there is some subset $T \subseteq S$ such that T contains exactly k elements and the sum of the elements in T is less than k^3, then your algorithm should output "no". Otherwise, it should output "yes". Argue (informally) that your algorithm is correct and analyze its running time. **[10 points]**

Problem 5. Finding the Missing Number [20 points]

Suppose you are given an unsorted array A of all integers in the range 0 to n except for one integer, denoted the *missing number*. Assume $n = 2^k - 1$.

(a) Design a $O(n)$ Divide and Conquer algorithm to find the missing number. Partial credit will be given for non Divide and Conquer algorithms. Argue (informally) that your algorithm is correct and analyze its running time. **[12 points]**

(b) Suppose the integers in A are stored as k-bit binary numbers, i.e. each bit is 0 or 1. For example, if $k = 2$ and the array $A = [01, 00, 11]$, then the missing number is 10.

Now the only operation to examine the integers is BIT-LOOKUP(i, j), which returns the jth bit of number $A[i]$ and costs unit time. Design an $O(n)$ algorithm to find the missing number. Argue (informally) that your algorithm is correct and analyze its running time. **[8 points]**

DETERMINISTICSELECT(A, n, i)
Divide the elements of the input array A into groups of 3 elements.
Find median of each group of 3 elements and put them in array B.
Call DETERMINISTICSELECT$(B, n/3, n/6)$ to find median of the medians, x.
Partition the input array around x into A_1 containing k elements $\leq x$
    and A_2 containing $n - k - 1$ elements $\geq x$.
If $i = k + 1$, then return x.
  Else if $i \leq k$, DETERMINISTICSELECT(A_1, k, i).
  Else if $i > k$, DETERMINISTICSELECT$(A_2, n - k - 1, i - (k + 1))$.