Practice Quiz 1

- The real Quiz 1 will be held on Thursday, October 15, in lecture.
- There will be a quiz review on Friday, October 9, during recitation.
- The quiz will be closed book. You may use one double sided Letter ($8\frac{1}{2}'' \times 11''$) or A4 crib sheet. No calculators or programmable devices are permitted.
- Write your solutions in the space provided. Extra scratch paper may be provided if you need more room, although your answer should fit in the given space.
- Do not waste time re-deriving facts that we have studied. It is sufficient to cite known results.
- Do not spend too much time on any one problem. Generally, a problem’s point value is an indication of how much time to spend on it.
- 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 1. Recurrences [15 points] (4 parts)
Solve the following recurrences by giving tight Θ-notation bounds. You do not need to justify your answers, but any justification that you provide will help when assigning partial credit. As usual, assume that for $n \leq 10$, $T(n) = O(1)$.

(a) [2 points] \[ T(n) = 3T(n/3) + 0.5n \lg(n). \]

(b) [4 points] \[ T(n) = 9T(\sqrt[3]{n}) + \Theta(\log(n)). \]
(c) [4 points] \[ T(n) = T(2n/7) + T(5n/7) + \Theta(n). \]

(d) [5 points] Define \( T(n) \) by the recursion \( T(n) = 9(T([n/3]) - 1) + n^3 + 2n \) for \( n \geq 1 \), with base case \( T(0) = 0 \). Prove \( T(n) \leq 3n^3/2 \).
**Problem 2. True or False, and Justify** [13 points] (6 parts)

Circle T or F for each of the following statements, and briefly explain why. Your justification is worth more points than your true-or-false designation.

(a) T F [2 points] To achieve asymptotically optimal performance, a skip list must use promotion probability $p = 0.5$.

(b) T F [2 points] Universal hashing requires that you know what elements you’ll hash in advance.
(c) T F  [2 points] In a B-tree, the maximum number of children of an internal non-root node is at most twice the minimum of number of children.

(d) T F  [2 points] A rotate operation on balanced tree always increases the depth of at least one node and decreases the depth of at least one node.

(e) T F  [3 points] In a B-tree of minimum parameter $t$, every node contains at least $t - 1$ elements.
Problem 3. Short Answer [13 points] (4 parts)

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

(a) [2 points] What is the expected difference between the depth of the deepest leaf and the depth of the least deep leaf in a 2-3-4 tree containing $N$ elements?

(b) [3 points] Show how to find a divisor $d$ of $N$ such that $d$ is not 1 or $N$, given $x, y$ such that $x^2 = y^2 \mod N$ and $x \neq y \mod N, x \neq -y \mod N.$
(c) [4 points] Let $\mathcal{H}$ be a universal hash family mapping $[1 \ldots N]$ to $[1 \ldots M]$. Let $X_{ijh}$ be the indicator variable for a collision between $i$ and $j$ under the hash function $h$, $i \neq j$ and $h \in \mathcal{H}$. What is $E(X_{ijh})$, where the expectation is taken over $i$, $j$, and $h$?

(d) [4 points] Consider a balanced binary tree of $n$ elements in which each node has an integer value. The weight of a path is the sum of the values of the nodes visited by the path. Give an optimal algorithm that computes the maximum possible weight of a path in the binary tree, starting at the root. What is the running time of your algorithm?
Problem 4. Slightly-Longer Short Answer [29 points] (5 parts)

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

(a) [5 points] A sequence of \( n \) operations is performed, so that the \( i \)th operation costs \( \lg(i) \) if \( i \) is an exact power of 2, and 1 otherwise. That is the amortized cost per operation?

(b) [6 points] Define set \( S = \{A \mid A = x^{(N-1)/2} \pmod{N}\} \) for a prime \( N \). Is \( S \) a sub-group of \( Z_N^* \)? If so, what can you say about the size of set \( S \)?
(c) [6 points] Given two sets $A$ and $B$ of $n$ integers, give an efficient deterministic algorithm to find $A \cap B$ and analyze its runtime. Can you do better with randomization? Explain.

(d) [6 points]
Consider an array $A$ of $n$ integers. Find all elements occurring at least $n/3$ times.
Consider a sorted array $A$ of size $n$, containing distinct integers. Give an $O(\lg n)$ algorithm to find an index $i$ such that $A[i] = i$ (or none, if no such index exists). Does your algorithm still work if $A$ contains repeat elements? Explain why or why not.
Problem 5. **Searching in multiple lists** [8 points]

Consider two disjoint sorted arrays $A[1 \ldots m]$ and $B[1 \ldots n]$. Find an $O(\log k)$ time algorithm for computing the $k$-th smallest element in the union of the two arrays.
**Problem 6. The Eccentric Landlord** [8 points] (2 parts)

Your construction firm is hired to build an apartment building for an eccentric landlord. He wants his building to be a square of size $M \times M$, containing $M^2$ identical square apartments.

The landlord will add one tenant a day. When he can’t fit a new tenant, he will tear down two sides of the building and have new walls built, expanding it an $(M + 1) \times (M + 1)$ building.

It costs your firm $1 to build one apartment’s exterior wall; your other costs (demolishing exterior walls, building interior walls, etc) are negligible. Your costs will be:

- **Day 1:** $4 (build four walls)
- **Day 2:** $6 (expand to 2x2)
- **Day 3:** $0 (tenant moves into empty unit)
- **Day 4:** $0 (tenant moves into empty unit)
- **Day 5:** $8 (expand to 3x3)
- **Day 6:** $0 (tenant moves into empty unit)

The costs incurred on day 2 are shown below.

(a) [4 points] What will your asymptotic aggregate cost be for this project? Give your answer as a function of the number of days elapsed.
(b) [4 points] You convince the landlord to expand his building in bigger steps: Whenever he can’t fit a new tenant, he will double the building side length (instead of increasing it by one unit). Repeat your analysis from part (a) for this new condition.
Problem 7. Chemical testing [15 points]

A chemistry lab is given \( n \) samples, with the goal of determining which of the samples contain traces of a foreign substance. It is assumed that only few (say, at most \( t \)) samples test positive. The tests are very sensitive, and can detect even the slightest trace of the substance in a sample. However, each test is very expensive. Because of that, the lab decided to test "sample pools" instead. Each pool contains a mixture of some of the samples (each sample can participate in several pools). A test of a pool returns positive if any of the samples contributing to the pool contains a trace of the substance.

Design a testing method that correctly determines the positive samples using only \( O(t \log n) \) tests. The method can be adaptive, i.e., the choice of the next test can depend on the outcomes of the previous tests.