Problem 1A (Recurrence Relations — 3 Styles, 9 pts)

Use induction to show the asymptotic run-time corresponding to the following recurrence relation is $O(\log^2(n))$. **Do not use the Master Theorem here.**

$$T(0) = T(1) = 1. \text{ For all } n \geq 2, T(n) = T(\lceil \tfrac{n}{2} \rceil - 1) + \log n.$$

Problem 1B (9 pts)

Use a recursion tree to find the asymptotic run-time corresponding to the following recurrence relation. You may assume any fractional input to T is the greatest integer less than it (e.g., $T(\frac{3n}{5}) = T(\lfloor \frac{3n}{5} \rfloor)$).

$$T(0) = T(1) = 1. \text{ For all } n \geq 2,\ T(n) = T(\tfrac{3n}{5}) + T(\tfrac{4n}{5}) + n^2$$

Problem 1C (12 pts)

Find the asymptotic run-time corresponding to each of the following recurrence relations using the Master Theorem. For each, explain which case of the Master Theorem applies and why.

(a) $T(n) = 3T(\frac{n}{8}) + \sqrt{2n}$.

(b) $T(n) = 3T(\frac{n}{3}) + 3n$.

(c) $T(n) = 4T(\frac{n}{2}) + n \log^2 n$.

Problem 2 (Array Search, 30 pts)

You are given an array of n integers $a_1 < a_2 < \cdots < a_n$. Give an $O(\log n)$ algorithm that outputs an index i where $a_i = i$, or outputs $\perp$ if such i does not exist. Justify the correctness and time complexity of your proposed algorithm.

Problem 3 (Malfunctioning Phones, 40 pts)

A manufacturer has a recall on a set of n cell phones, some of which have a malfunction which makes them unreliable. The manufacturer has built a machine that allows a pair of phones to test each other's correctness. Let C_1, C_2 be a pair of phones. The machine M runs in the following way:

1. $M(C_1, C_2) = 11$ if both phones say the other is working.
2. $M(C_1, C_2) = 10$ if one phone says the other is working and one phone says the other is malfunctioning.
3. $M(C_1, C_2) = 00$ if both phones say the other is malfunctioning.

Remember that malfunctioning phones cannot be trusted, so they may lie, tell the truth, or throw out a random response. Working phones, on the other hand, can be assumed to know if the other phone is working or malfunctioning always.

(a) Show that if you know at least one working phone, all other working phones can be found by using $O(n)$ queries to M.

(b) Assume the majority of the phones are working, i.e., there are greater than $n/2$ working phones. Give an algorithm that can find a working phone in $O(n)$ queries to M. Justify the correctness and time complexity of your proposed algorithm. [Hint: Start by explaining how to use $O(n)$ queries to reduce the problem size by a constant factor.]

(c) Assume the majority of the phones are malfunctioning, i.e., there are fewer than $n/2$ working phones. Is there still a procedure (using M) that is guaranteed to find a working phone? Give a brief justification.