1. **Ternary NAND trees [20 points].** Consider the problem where \( n = 3^k \) and the goal is to evaluate a (balanced) ternary NAND tree on inputs \( x_1, \ldots, x_n \). A ternary NAND tree on inputs \( x_1, \ldots, x_n \) is a formula defined recursively as

\[
f_0(x_1) = x_1 \\
f_{k+1}(x_1, \ldots, x_{3^k+1}) = \text{NAND}(f_k(x_1, \ldots, x_{3^k}), f_k(x_{3^k+1}, \ldots, x_{2\cdot3^k}), f_k(x_{2\cdot3^k+1}, \ldots, x_{3\cdot3^k})).
\]

(Note that \( \text{NAND}(a, b, c) = a \land b \land c \).)

Give the most efficient randomized algorithm that you can for this problem. Your algorithm should solve the problem with an expected number of queries that is \( O(n^d) \), and your goal is to make \( d \) as small as possible. Include a clear analysis of the expected number of queries of your algorithm. Also, include an exact expression for the exponent \( d \) as well as a numerical approximation of \( d \). (Hint: consider a recursive algorithm that computes \( f_{k+1}(x_1, \ldots, x_{3^k+1}) \) somehow in terms of \( f_k(x_1, \ldots, x_{3^k}), f_k(x_{3^k+1}, \ldots, x_{2\cdot3^k}), \) and \( f_k(x_{2\cdot3^k+1}, \ldots, x_{3\cdot3^k}) \).)

2. **Communication complexity of GREATER-THAN function [15 points].** The GREATER-THAN (GT) function is defined as

\[
\text{GT} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}
\]

where, for \( x, y \in \{0, 1\}^n \),

\[
\text{GT}(x, y) = \begin{cases} 
1 & \text{if } x > y \\
0 & \text{otherwise.}
\end{cases}
\]

For the definition of \( > \), we interpret \( \{0, 1\}^n = \{0, 1, \ldots, 2^n - 1\} \) in the usual way that numbers are denoted in binary.

(a) [5 points] Show that the deterministic communication complexity of GT is \( \Omega(n) \). For this question, you may assume that the deterministic communication complexity of EQUALITY (EQ) is at least \( n \) (as was shown in class).

(b) [10 points] Show that the randomized communication complexity of GT is \( O(\log^2 n) \), where the allowed error probability is (say) \( \frac{1}{4} \).

For this question, you may assume that there is a randomized \( O(\log(n) + \log(1/\epsilon)) \) communication protocol for EQUALITY (EQ) that errs with probability at most \( \epsilon \). You may use this communication protocol for EQ as a subroutine in your protocol, with various substrings of \( x \) and \( y \), in order to solve GT.
3. **Schönig’s algorithm when there are multiple satisfying assignments [10 points].**

Recall that in class, in our analysis of Schönig’s algorithm for 3SAT, we assumed that the instance of 3SAT $f$ has a unique satisfying assignment $x^*$. The case where $f$ has more than one satisfying assignment was not explained. There were vague comments that the case of a unique satisfying assignment is the “hardest case”, and assertions that the algorithm will perform at least as well if there are more satisfying assignments. Let us analyze this part more rigorously.

Consider the main step in Schönig’s algorithm: for truth assignment $x$, take an unsatisfied clause of $f$ and flip one of the three bits of $x$ corresponding to the variables in that clause with uniform probability.

We argued in class that, in the case of a unique satisfying assignment $x^*$, if the Hamming distance between $x$ and $x^*$ is $u$ then, after $u$ iterations of the above step, the probability of reaching $x^*$ is at least $(\frac{1}{3})^u$. (This was for the simpler upper bound.)

Now, let $f$ be an instance of 3SAT with $n$ variables that has (say) two different satisfying assignments, $x^*$ and $x^{**}$. Let $u$ be the Hamming distance between $x$ and $x^*$.

(a) [5 points] Is it still the case that, after $u$ iterations of the above step, the probability of reaching $x^*$ (not $x^{**}$) is at least $(\frac{1}{3})^u$? Explain your answer.

(b) [5 points] Explain why, after $u$ iterations of the above step, the probability of reaching either $x^*$ or $x^{**}$ is at least $(\frac{1}{3})^u$.

4. **Schönig’s algorithm for 4SAT [15 points].**

Let 4SAT be the satisfiability problem for formulas in 4CNF form (where each clause is the OR of four literals).

Consider the main procedure in Schönig’s algorithm for this problem: start with a random truth assignment $x_{\text{init}}$ and repeat the step of taking an unsatisfied clause of $f$ and flipping one of the four bits of $x$ corresponding to the variables in that clause with uniform probability.

Let us assume that our instance of 4SAT, $f$, has a unique satisfying assignment $x^*$.

(a) [5 points] If the Hamming distance between $x_{\text{init}}$ and $x^*$ is $u$, what is the probability that $x^*$ is reached within $u$ steps?

(b) [5 points] If the Hamming distance between $x_{\text{init}}$ and $x^*$ is $u$, what is the probability that $x^*$ is reached within $3u$ steps?

(c) [5 points] Give the best upper bound that you can on the exponent for solving 4SAT using Schönig’s algorithm. That is, the minimum $\alpha > 0$ such that the running time is $O(2^{\alpha n})$. 


