1. (10 pts) Consider the following recurrence relation:

\[
T(n) = \begin{cases} 
5 & n = 1 \\
T(\left\lfloor \frac{n}{2} \right\rfloor) * T(\left\lceil \frac{n}{2} \right\rceil) & n > 1
\end{cases}
\]

Use repeated substitution (aka unwinding) to make a conjecture of a closed-form expression for \( T(n) \) in the special case where \( n \) is a power of 2 (i.e., \( \exists k \in \mathbb{N}, n = 2^k \)). Then, prove your conjecture is true for \( n \) of the form \( 2^k \).

**Sample Solution.**

When \( n > 1 \)

\[
T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) * T\left(\left\lceil \frac{n}{2} \right\rceil\right) = T\left(\frac{n}{2}\right) * T\left(\frac{n}{2}\right) = T^2\left(\frac{n}{2}\right) \quad \# \text{since } n = 2^k
\]

\[
= T^4\left(\frac{n}{4}\right)
\]

\[
= T^8\left(\frac{n}{8}\right)
\]

\[
\ldots
\]

\[
= T^{2^k}\left(\frac{n}{2^k}\right)
\]

\[
= T^n(1)
\]

\[
= 5^n
\]

**Proof by simple induction on \( k \).**

**Basis step.** \( k = 0, n = 2^0 = 1. T(1) = 5 = 5^1. \)

**Inductive step.** Assume \( P(k) \) holds for an arbitrary \( k \in \mathbb{N} \). That means

\[
T\left(2^k\right) = T^2\left(\frac{2^k}{2}\right) = T^2\left(2^{k-1}\right) = 5^{2^k}
\]

By using IH, we must show \( T\left(2^{k+1}\right) = 5^{(2^k+1)} \).

\[
T\left(2^{k+1}\right) = T^2\left(\frac{2^{k+1}}{2}\right)
\]

\[
= T^2\left(2^k\right)
\]

\[
= (5^{2^k})^2 \quad \# \text{by IH.}
\]

\[
= 5^{(2^k)\times 2} = 5^{(2^k+1)}
\]
2. (8 pts) Assume we know that when \( n = 2^{(2^k)} \) for some \( k \in \mathbb{N} \), \( S(n) = \lg \lg n + 3 \). Show that \( S(n) \) is in \( O(\lg \lg n) \) for all \( n \geq B \in \mathbb{N} \), not just special cases, and find a value for \( B \).

**Hint:** you may use

i. \( S \) is monotonic non-decreasing

ii. \( \forall n > 1 \in \mathbb{N}, \exists k \in \mathbb{N} \) such that \( \sqrt{2^{(2^k)}} \leq n \leq 2^{(2^k)} \)

iii. \( \sqrt{2^{(2^k)}} = 2^{(2^k-k)} \)

iv. Since \( n = 2^{(2^k)}, 2^k = \lg n \) and \( k = \lg \lg n \)

**Note that we want to show \( S(n) \) for all \( n \geq B \in \mathbb{N} \) is in \( O(\lg \lg n) \).**

**Sample Solution 1.**
Assume \( \sqrt{2^{2^k}} \leq n \leq 2^{2^k} \)

\[
S(n) \leq S \left( 2^{2^k} \right) \quad \text{# since } S \text{ is non-decreasing}
\]

\[
S(n) \leq \lg \lg 2^{2^k} + 3
\]

\[
S(n) \leq \lg \lg n^2 + 3 \quad \text{# since } \sqrt{2^{2^k}} \leq n, \text{ then } 2^{2^k} \leq n^2
\]

\[
S(n) \leq \lg (2 \lg n) + 3
\]

\[
S(n) \leq 1 + \lg \lg n + 3
\]

\[
S(n) \leq \lg \lg n + 4
\]

\[
S(n) \leq \lg \lg n + 4 \lg \lg n \quad \text{# when } n \geq 4
\]

\[
S(n) \leq 5 \lg \lg n
\]

\( C = 5 \) and \( B = 4 \)

**Sample Solution 2.**
Assume \( \sqrt{2^{2^k}} \leq n \leq 2^{2^k} \)

\[
S(n) \leq S \left( 2^{2^k} \right) \quad \text{# since } S \text{ is non-decreasing}
\]

\[
S(n) \leq \lg \lg 2^{2^k} + 3
\]

\[
S(n) \leq k + 3
\]

\[
S(n) \leq (k - 1) + 4
\]
\[ S(n) \leq \log \log 2^{(2^k-1)} + 4 \]
\[ S(n) \leq \log \log \sqrt[2k]{2} + 4 \]
\[ S(n) \leq \log \log n + 4 \quad \text{# since } \log \text{ is monotonic} \]
\[ S(n) \leq \log \log n + 4 \log \log n \quad \text{# when } n \geq 4 \]
\[ S(n) \leq 5 \log \log n \]

C= 5 and B=4
Note. This question may take more time than the number of points assigned suggest.

3.

a) (4 pts) Consider the following algorithm, and prove if the loop iterates at least $c$ times, the following loop invariant holds at end of the $c$-th iteration.

$$LI(i_c, \text{sum}_c) : 0 \leq i_c \leq \frac{n}{3}, i_c \in \mathbb{N} \text{ and sum}_c = \sum_{j=0}^{i_c-1} A[3j]$$

Note that sum of an empty list is zero, i.e., $\sum_{j=0}^{-1} A[3j] = 0$.

1. **Algorithm** avg(A)
   **pre**: A is a list of real numbers, its index starts from 0 and its size, $n$, is $3k$, $\exists k > 0 \in \mathbb{N}$
   **post**: return the average of numbers in positions divisible by 3
2. $i = 0$
3. $\text{sum} = 0$
4. $m = \text{length}(A)/3$
5. **while** $i < m$
6. $\text{sum} = \text{sum} + A[3 \ast i]$
7. $i = i + 1$
8. $a = \text{sum} / i$
9. **return** $a$

b) (1 pts) Partial correctness: use the loop invariant above and prove the algorithm is correct, assuming it terminates.

a)

Sample solution.

**Proof by simple induction.**

Let $P(c)$ denote if the loop iterates at least $c$ times, then $0 \leq i_c \leq \frac{n}{3}, i_c \in \mathbb{N}$ and $\text{sum}_c = \sum_{j=0}^{i_c-1} A[3j]$.

**Basis step.**

$P(0)$ holds since if the loop iterates at least 0 times, i.e. before entering the loop:

$$0 \leq i_0 = 0 \leq \frac{n}{3}, i_0 = 0 \in \mathbb{N} \text{ and}$$

$$\text{Sum}_0 = 0 = \sum_{j=0}^{i_0-1} A[3j] = \sum_{j=0}^{-1} A[3j] = 0$$
Inductive step.

Assume \( P(k) \) holds, i.e., if the loop iterates at least \( k \) times, then \( i_k \in \mathbb{N}, 0 \leq i_k \leq \frac{n}{3} \) and \( \text{Sum}_k = \sum_{j=0}^{i_k-1} A[3j] \). We must show \( P(k) \to P(k + 1) \).

**case 1:** if the loop does not iterate \( k + 1 \) times, \( P(k + 1) \) is vacuously true.

**case 2:** If the loop iterates at least \( k + 1 \) times,

\[
0 \leq i_k < m = \frac{n}{3} \quad \# \text{by Line 5}
\]

Then,

\[
\text{Sum}_{k+1} = \text{Sum}_k + A[3 \times i_k] \quad \# \text{at Line 6}
\]

\[
= \sum_{j=0}^{i_k-1} A[3j] + A[3 \times i_k] \# \text{by IH}
\]

\[
= \sum_{j=0}^{i_{k+1}-1} A[3j] \quad \# \text{since } i_k = i_{k+1} - 1
\]

Also,

\[
i_{k+1} = i_k + 1 \quad \# \text{Line 7}
\]

\[
0 \leq i_{k+1} \leq \frac{n}{3} \quad \# \text{since } 0 \leq i_k < \frac{n}{3}, i_k \in \mathbb{N}
\]

This completes the inductive step, as \( P(k + 1) \) holds.

Hence, if the loop iterates at least \( c \) times, the following loop invariant holds at end of the \( c \)-th iteration.

**b)**

**Sample Solution.**

Since the loop terminates and by LI \( 0 \leq i_c \leq \frac{n}{3} \) at end of iteration \( c \), then \( i_c = m = \frac{n}{3} \) when the loop exits. Also, by LI, when the loop exits:

\[
\text{Sum}_{\frac{n}{3}} = \sum_{j=0}^{\frac{n}{3}-1} A[3j] \quad \text{which is sum of elements of } A \text{ at positions divisible by 3, up to and including position } n - 3.
\]

The number of elements at positions divisible by 3 is \( \frac{n}{3} = i_n \). The program returns \( \text{Sum}_{\frac{n}{3}}/i_n \) which is the average by definition.

Hence, \( \text{preconditions } \to \text{postconditions} \)