Opening Questions

1. Define Big-O, Omega and Theta notation
2. In your own words explain what a recurrence relation is, what do we use recurrence relations for, why do we solve recurrence relations?

Recurrence Relation Solution Approach—Guess and prove by induction

Guess (or are given Hint) at form of solution, probe it is the solution

- Using definition of BIG-O or θ
- Using induction
  - Probe Base Case (if boundary condition given)
  - Assume true for some n
  - Prove true for a larger n

Example:

$$T(n) = 4T(n/2) + 2 \text{ guess } T(n) = O(n^3) \text{ ??}$$

Assume $T(k) \leq ck^3$ for some $k < n$, use assumption with $k = n/2$, then prove it for $k = n$

$T(n/2) \leq c(n/2)^3$ merge with recurrence

$$\begin{aligned}
&T(n) \leq 4c(n/2)^3 + n \\
&T(n) \leq \frac{c}{2n^3} + n \\
&T(n) \leq cn^3 - \left(\frac{c}{2n^3} - n\right) \\
&\quad\left(\frac{c}{2n^3} - n\right) > 0 \text{ if } c \geq 2 \text{ and } n > 1 \\
&T(n) \leq cn^3 - (\text{something positive}) \\
&T(n) \leq cn^3 \\
&T(n) = O(n^3)
\end{aligned}$$

1. $$T(n) = 4T(n/2) + n^3 \text{ guess } T(n) = \Theta(n^3)\text{??}$$

2. $$T(n) = 4T(n/2) + n \text{ guess } T(n) = O(n^2)\text{??}$$

Recurrence Relation Solution Approach - Iteration Method (repeated substitution)

Convert the recurrence relation to summation using repeated substitution (Iterations)

- Keys to Iteration Method

  - # of times iterated to get $T(1)$
  - Find the pattern in terms and simplify to summation

EXAMPLE

$$
\begin{aligned}
T(n) &= 4T\left(\frac{n}{2}\right) + n \\
T(n) &= n + 4T\left(\frac{n}{2}\right) \\
&= n + 4\left(4T\left(\frac{n}{4}\right) + \frac{n}{2}\right) \\
&= n + 4\left(\frac{n}{2} + 4\left(\frac{n}{4} + 4T\left(\frac{n}{8}\right)\right)\right) \\
&= (n + 2n + 4n + 64)T\left(\frac{n}{8}\right) \\
T(n) &= n + 2n + 4n + \cdots + 4^{\lg_2(n)}T(1) \\
&= n\sum_{k=0}^{\lg(n-1)} 2^k + \theta(n^2) \\
&= n\left(\frac{2^{\lg(n)} - 1}{2 - 1}\right) \\
T(n) &= n^2 - n \\
&= O(n^2)
\end{aligned}
$$

3. $$T(n) = 3T(n/3) + \lg n \text{ proof by iteration/repeated substitution}$$

4. $$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

5. $$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

6. $$T(n) = T(n-1) + n$$

Recurrence Relation Solution Approach – Master Method

For solving recurrences of the form $T(n) = aT\left(\frac{n}{b}\right) + f(n)$

- Compare growth of $f(n)$ to $n^{\log_b(a)}$

Case 1)

$$f(n) \leq cn^{\log_b(a)} \; T(n) = \Theta(n^{\log_b(a)})$$

Case 2)

$$c_1 n^{\log_b(a)} \leq f(n) \leq c_2 n^{\log_b(a)} \; T(n) = \Theta(n^{\log_b(a)} \lg(n))$$

Case 3)

$$f(n) \geq cn^{\log_b(a)} \; T(n) = \Theta(f(n))$$

7.
$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

8.
$$T(n) = 7T\left(\frac{n}{2}\right) + O(n^2)$$

9.
$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$