DESIGN AND ANALYSIS OF ALGORITHMS

CS 4120/5120

GUESS-AND-VERIFICATION

AGENDA

- Three methods to solve recurrence
  - Guess-and-verification
  - Recursion tree
  - Master theorem

GUESS-AND-VERIFICATION

- Key steps
  - **Guess** the bound of the given function.
  - **Verify** the bound using mathematical induction.

MAKING A GOOD GUESS #1 USING EMPIRICAL BOUNDS

- Goal: Bound function f.

- Scenario **#1**
  - The function f is **in the same form as function g**.
  - Conclude the bound of function f is the same as that of g.

MAKING A GOOD GUESS #1 USING EMPIRICAL BOUNDS

- Goal: Bound function f.

- Scenario **#2**
  - The function f is **close in form to function** g.
  - Guess the bound of function f is the same as that of g.

MAKING A GOOD GUESS #2 USING EMPIRICAL BOUNDS

- Goal: Bound function f.

- Have no clue of the bound
  - Method 1
    - **Backward** substitution to derive a closed-end form of the function in a ***top-down*** manner.
  - Method 2
    - **Forward** substitution to derive a close-end form of the function in a ***bottom-up*** manner.

Backward Substitution to Make a Good Guess

- **Guess** the bound of recurrence $T(n) = \begin{cases} \boldsymbol{\Theta(1)}, & \boldsymbol{n = 1} \\ \mathbf{3} \cdot T\left(\frac{\boldsymbol{n}}{\mathbf{4}}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.

- Observe the recurrence
  - Parameter n is **divided by 4** in new recursion.
  - Each recursion has **three repetitions** of the same function with $n/4$ being the parameter.
  - We are given a **bottoms-out case**.

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{\boldsymbol{n}}{\boldsymbol{4}}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.

- Solution

  – **Step 1**: Repeatedly **substitute** the parameter by $\frac{1}{4}$ of parameter to **expand** the function k times.

$$T(n) = 3 \cdot T\left(\frac{\boldsymbol{n}}{\boldsymbol{4}}\right) + n$$

$$= 3 \cdot \left(\boldsymbol{3 \cdot T\left(\frac{n}{4^2}\right) + \frac{n}{4^1}}\right) + n = \underline{3^2 T\left(\frac{n}{4^2}\right) + \left(\frac{3}{4} + 1\right) n}$$

$$= \underline{3^3 T\left(\frac{n}{4^3}\right) + \left(\left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right) n}$$

(**1**st expansion: $T\left(\frac{\boldsymbol{n}}{\boldsymbol{4}}\right) = 3 \cdot T\left(\frac{n}{4^2}\right) + \frac{n}{4^1}$)

(**2**nd expansion: $T\left(\frac{n}{4^2}\right) = 3 \cdot T\left(\frac{n}{4^3}\right) + \frac{n}{4^2}$)

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.

- Solution

  – **Step 1**: Repeatedly **substitute** the parameter by $\frac{1}{4}$ of parameter to **expand** the function k times.

$$T(n) = 3^3 T\left(\frac{n}{4^3}\right) + \left(\left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right)n = 3^3 T\left(3 \cdot T\left(\frac{n}{4^4}\right) + \frac{n}{4^3}\right) + \left(\left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right)n$$

$$= 3^4 T\left(\frac{n}{4^4}\right) + \left(\left(\frac{3}{4}\right)^3 + \left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right)n$$

(3rd expansion: $T\left(\frac{n}{4^3}\right) = 3 \cdot T\left(\frac{n}{4^4}\right) + \frac{n}{4^3}$)

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 1**: Repeatedly **substitute** the parameter by $\frac{1}{4}$ of parameter to **expand** the function k times.

$$T(n) = 3^4 T\left(\frac{n}{4^4}\right) + \left(\left(\frac{3}{4}\right)^3 + \left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right)n = 3^4 T\left(3 \cdot T\left(\frac{n}{4^5}\right) + \frac{n}{4^4}\right) + \left(\left(\frac{3}{4}\right)^3 + \left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right)n$$

$$= 3^5 T\left(\frac{n}{4^5}\right) + \left(\left(\frac{3}{4}\right)^4 + \left(\frac{3}{4}\right)^3 + \left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right)n$$

(4th expansion: $T\left(\frac{n}{4^4}\right) = 3 \cdot T\left(\frac{n}{4^5}\right) + \frac{n}{4^4}$)

What would the kth expansion (using the kth recurrence) look like?

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 1**: Repeatedly **substitute** the parameter by $\frac{1}{4}$ of parameter to **expand** the function k times.

$$T(n) = 3^5 T\left(\frac{n}{4^5}\right) + \left(\left(\frac{3}{4}\right)^4 + \left(\frac{3}{4}\right)^3 + \left(\frac{3}{4}\right)^2 + \frac{3}{4} + 1\right) n$$

(4th expansion: $T\left(\frac{n}{4^4}\right) = 3 \cdot T\left(\frac{n}{4^5}\right) + \frac{n}{4^4}$)

$$= \cdots$$

$$= 3^{k+1} T\left(\frac{n}{4^{k+1}}\right) + \left(\left(\frac{3}{4}\right)^k + \left(\frac{3}{4}\right)^{k-1} + \cdots + \left(\frac{3}{4}\right)^1 + \left(\frac{3}{4}\right)^0\right) n$$

(kth expansion: $T\left(\frac{n}{4^k}\right) = 3 \cdot T\left(\frac{n}{4^{k+1}}\right) + \frac{n}{4^k}$)

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.

- Solution
  - **Step 1**: Repeatedly **substitute** the parameter by $\frac{1}{4}$ of parameter to **expand** the function k times.

$$T(n) = 3^{k+1} T\left(\frac{n}{4^{k+1}}\right) + \left(\boxed{\left(\frac{3}{4}\right)^{k} + \left(\frac{3}{4}\right)^{k-1} + \cdots + \left(\frac{3}{4}\right)^{1} + \left(\frac{3}{4}\right)^{0}}\right) n$$

(kth expansion: $T\left(\frac{n}{4^k}\right) = 3 \cdot T\left(\frac{n}{4^{k+1}}\right) + \frac{n}{4^k}$)

$$\| $$

$$\left(\frac{3}{4}\right)^{0} \cdot \frac{1 - \left(\frac{3}{4}\right)^{k+1}}{1 - \frac{3}{4}} = 4\left(1 - \left(\frac{3}{4}\right)^{k+1}\right)$$

Summation of **the first $k+1$ terms** of a geometric series with the first term being $(3/4)^0$ and the constant ratio being $3/4$.

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.

- Solution

  – **Step 1**: Repeatedly **substitute** the parameter by $\frac{1}{4}$ of parameter to **expand** the function k times.

$$T(n) = 3^{k+1} T\left(\frac{n}{4^{k+1}}\right) + \left(\qquad 4\left(1 - \left(\frac{3}{4}\right)^{k+1}\right) \qquad \right) n$$

(kth expansion: $T\left(\frac{n}{4^k}\right) = 3 \cdot T\left(\frac{n}{4^{k+1}}\right) + \frac{n}{4^k}$)

$$= 3^{k+1} T\left(\frac{n}{4^{k+1}}\right) + \left(4\left(1 - \left(\frac{3}{4}\right)^{k+1}\right)\right) n$$

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 2

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{\boldsymbol{n}}{\boldsymbol{4}}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 2**: Let the kth recurrence be the one that will recurse on the bottoms-out case.

    $\boldsymbol{k}$th recurrence: $T\left(\frac{n}{4^{\boldsymbol{k}}}\right) = 3 \cdot T\left(\frac{n}{4^{\boldsymbol{k+1}}}\right) + \frac{n}{4^{\boldsymbol{k}}}$

    - Let $\frac{\boldsymbol{n}}{4^{\boldsymbol{k+1}}} \boldsymbol{= 1}$.

    $$T\left(\frac{n}{4^{\boldsymbol{k}}}\right) = 3 \cdot T\left(\frac{n}{4^{\boldsymbol{k+1}}}\right) + \frac{n}{4^{\boldsymbol{k}}} \Longrightarrow T\left(4\right) = 3 \cdot T\left(1\right) + 4 \quad \Longrightarrow T(4) = \boldsymbol{\Theta(1)}$$

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 3

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{\boldsymbol{n}}{\boldsymbol{4}}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  – **Step 3**: Solve $\frac{n}{4^{\boldsymbol{k}+\mathbf{1}}} = \mathbf{1}$ for $\boldsymbol{k}$.

$\Longrightarrow n = 4^{\boldsymbol{k}+\mathbf{1}}$ (Multiply by $4^{\boldsymbol{k}+\mathbf{1}}$)

$\Longrightarrow \lg n = \lg\left(4^{\boldsymbol{k}+\mathbf{1}}\right)$ (Take $\lg$ of both sides $4^{\boldsymbol{k}+\mathbf{1}}$)

$\Longrightarrow \lg n = \lg\left((2^2)^{\boldsymbol{k}+\mathbf{1}}\right) = \lg 2^{2(\boldsymbol{k}+\mathbf{1})}$ ($\left(a^b\right)^c = a^{bc}$)

$\Longrightarrow \lg n = 2(\boldsymbol{k} + \mathbf{1})$ ($\log_a a^b = b$)

$\Longrightarrow \frac{\lg n}{2} - 1 = \boldsymbol{k}$

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 4

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 4**: Plug k back in the function after k-th expansion. $\frac{n}{4^{k+1}} = 1 \Rightarrow k + 1 = \log_4 n \Rightarrow k = \frac{\lg n}{2} - 1$

$$T(n) = 3^{k+1} T\left(\frac{n}{4^{k+1}}\right) + \left(4\left(1 - \left(\frac{3}{4}\right)^{k+1}\right)\right) n$$

(kth expansion: $T\left(\frac{n}{4^k}\right) = 3 \cdot T\left(\frac{n}{4^{k+1}}\right) + \frac{n}{4^k}$)

$$= \underline{3^{\log_4 n} \quad \Theta(1) \quad + 4\left(1 - \left(\frac{3}{4}\right)^{\log_4 n}\right)} n$$

BACKWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 4

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 4**: Plug k back in the function after k-th expansion. $\frac{n}{4^{k+1}} = 1 \Rightarrow k + 1 = \log_4 n \Rightarrow k = \frac{\lg n}{2} - 1$

$$T(n) = 3^{\log_4 n}\Theta(1) + 4\left(1 - \left(\frac{3}{4}\right)^{\log_4 n}\right)n = \Theta\left(n^{\log_4 3}\right) + 4n - 4 \cdot n \cdot n^{\log_4 3/4} \quad (a^{\log_b c} = c^{\log_b a})$$

$$= \Theta\left(n^{\log_4 3}\right) + 4n - 4 \cdot n^{1+\log_4 3/4} \quad (a^b \cdot a^c = a^{b+c})$$

$$= \underline{\Theta\left(n^{\log_4 3}\right) + 4n - 4 \cdot n^{\log_4 3}} \quad (1 = \log_a a, \log_a b + \log_a c = \log_a(bc))$$

$$= O(\underline{n})$$

BACKWARD SUBSTITUTION REVIEW

- **Step 1**: Start with $\boldsymbol{T(n)}$. Repeatedly **expand** the k times.
  - Show at least 3 expansions before showing the kth expansion.
- **Step 2**: Let the kth recurrence be the recurrence that will recurse on the bottoms-out case.
  - Obtain an equation regarding n and k.
- **Step 3**: Solve the equation for k.
- **Step 4**: Plug k back in the function after k-th expansion. **Guess** the bound.
- There is a version with blanks at the end of the slides. You may use it for practice.
  - Answers to Canvas questions must be in the same format.

BACKWARD SUBSTITUTION FREQUENTLY USED FORMULAS

- Summation of the first x terms of an arithmetic sequence
  - a_1 and a_x being the first and x-th term, respectively.

$$\sum_{1}^{x} a_x = \frac{(a_1 + a_x) \cdot x}{2}$$

- Summation of the first x terms of a geometric sequence
  - a_1 is the first term, r is the constant ratio.

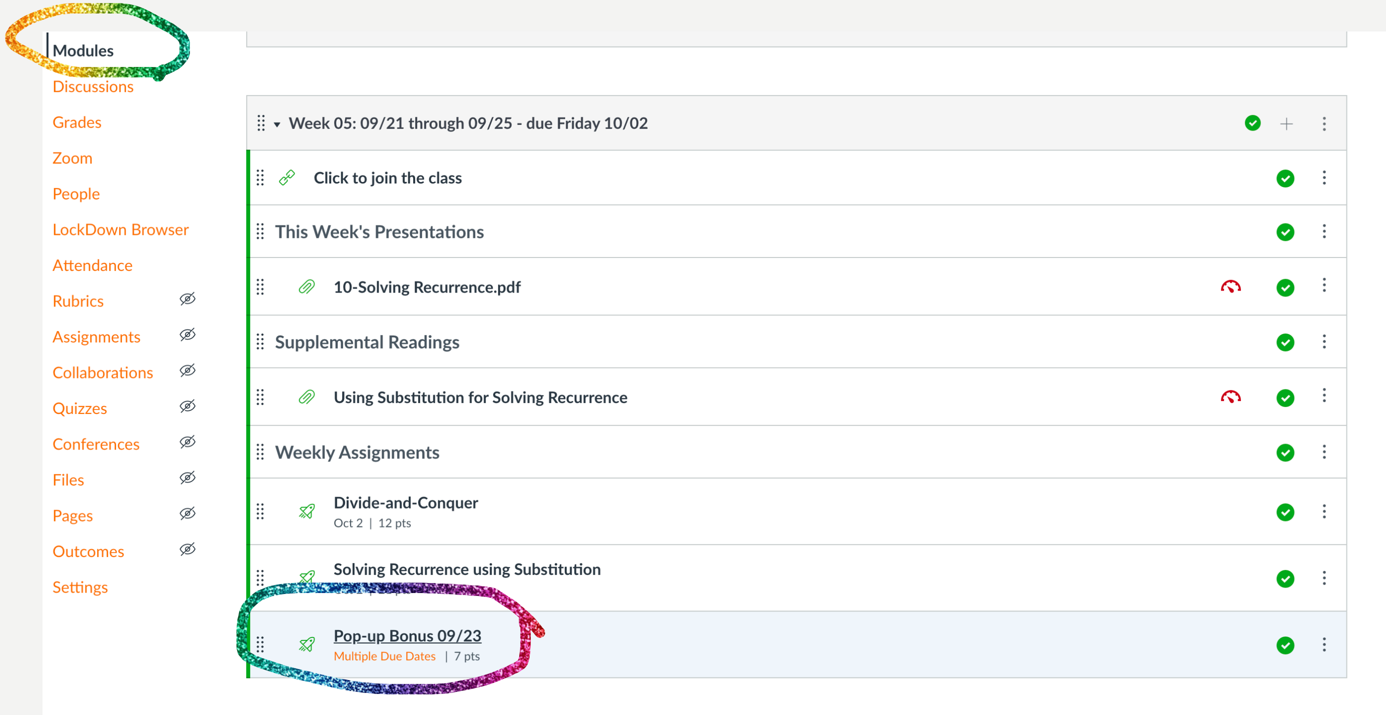
$$\sum_{1}^{x} a_x = \frac{a_1(1 - r^x)}{1 - r}$$

BACKWARD SUBSTITUTION
FREQUENTLY USED FORMULAS

- Logarithm identities
  - $a = b^{\log_b a}$
  - $\log_c(ab) = \log_c a + \log_c b$
  - $\log_b a^n = n \log_b a$
  - $\log_b(1/a) = -\log_b a$
  - $\log_b a = \frac{\log_c a}{\log_c b}, \log_b a = \frac{1}{\log_a b}$
  - $a^{\log_b c} = c^{\log_b a}$
- Exponential identities
  - $a^0 = 1$
  - $a^1 = a$
  - $a^{-1} = 1/a$
  - $(a^m)^n = a^{mn}$
  - $(a^n)^m = a^{nm}$
  - $a^m a^n = a^{m+n}$

[BONUS] 5-MIN CHALLENGE

- 7 bonus points
- Starts
  - Section 1001/5001 @ 2:30pm
  - Section 1002/5002 @ 3:30pm
- Ends in 5 minutes
- One attempt



BACKWARD SUBSTITUTION PRACTICE

- **Guess** the bound of recurrence $T(n) = \begin{cases} 1, & n = 1 \\ 2 \cdot T\left(\frac{n}{2}\right) + 1, & n > 1 \end{cases}$ using **backward** substitution.

- **Step 1**:

$T(n) = $ ____________________

$=$ ______________________________ (**1**st expansion:)

$=$ ______________________________ (**2**nd expansion:)

$=$ ______________________________ (**3**rd expansion:)

$= \ldots$

$=$ ______________________________ (***k***th expansion:)

BACKWARD SUBSTITUTION PRACTICE

- **Guess** the bound of recurrence $T(n) = \begin{cases} 1, & n = 1 \\ 2 \cdot T\left(\frac{n}{2}\right) + 1, & n > 1 \end{cases}$ using **backward** substitution.
- **Step 2**: Let ***k***th recurrence: ________________________ $= \boldsymbol{T(1)}$.
- **Step 3**: Solve for $\boldsymbol{k}$. _____________________________.
- **Step 4**:
  $T(n) =$ _______________________(fill the k-th expansion)

  $=$ __________________________
- **Guess** $T(n) =$ ______________________

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS

- **Guess** the bound of recurrence $T(n) = \begin{cases} \boldsymbol{\Theta(1)}, & \boldsymbol{n = 1} \\ \boldsymbol{3} \cdot T\left(\boldsymbol{\frac{n}{4}}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Observe the recurrence
  - We are given a **bottoms-out case**.
  - Based on the recurrence, parameter of the previous recurrence is **4 times the current parameter**.

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences by substituting the parameter by **4 times the parameter**.

1st substitution: $T(4) = 3 \cdot T(1) + 4 = \underline{3 \cdot c + 4}$

2nd substitution: $T(4^2) = 3 \cdot T(4) + 4^2 = 3 \cdot (3 \cdot c + 4) + 4^2 = \underline{3^2 c + 3 \cdot 4 + 4^2}$

3rd substitution: $T(4^3) = 3 \cdot T(4^2) + 4^3 = 3 \cdot (3^2 c + 3 \cdot 4 + 4^2) + 4^3 = \underline{3^3 c + 3^2 \cdot 4 + 3 \cdot 4^2 + 4^3}$

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences by substituting the parameter by **4 times the parameter**.

3rd substitution: $T(4^3) = 3 \cdot T(4^2) + 4^3 = 3^3 c + (3^2 \cdot 4 + 3 \cdot 4^2 + 4^3) = 3^3 c + \sum_{i=1}^{3} 3^{3-i} \cdot 4^i$

4th substitution: $T(4^4) = 3 \cdot T(4^3) + 4^4 = 3 \cdot \left(3^3 c + \sum_{1}^{3} 3^{3-i} \cdot 4^i\right) + 4^4 = 3^4 c + \sum_{i=1}^{4} 3^{4-i} \cdot 4^i$

What would the ***k***th substitution look like?

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \boldsymbol{\Theta(1)}, & \boldsymbol{n = 1} \\ \mathbf{3} \cdot T\left(\frac{\boldsymbol{n}}{\mathbf{4}}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences by substituting the parameter by **4 times the parameter**.

4th substitution: $T(\mathbf{4^4}) = \mathbf{3} \cdot T(\mathbf{4^3}) + \mathbf{4^4} = \mathbf{3^4}\boldsymbol{c} + \sum_{\boldsymbol{i=1}}^{\mathbf{4}} \mathbf{3}^{\mathbf{4}-\boldsymbol{i}} \cdot \mathbf{4}^{\boldsymbol{i}}$

$\boldsymbol{k}$th substitution: $T(\mathbf{4}^{\boldsymbol{k}}) = \mathbf{3} \cdot T(\mathbf{4}^{\boldsymbol{k-1}}) + \mathbf{4}^{\boldsymbol{k}} = \mathbf{3}^{\boldsymbol{k}}\boldsymbol{c} + \sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \mathbf{3}^{\boldsymbol{k-i}} \cdot \mathbf{4}^{\boldsymbol{i}}$

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences by substituting the parameter by **4 times the parameter**.

kth substitution: $T(4^k) = 3^k c + \sum_{1}^{k} 3^{k-i} \cdot 4^i = 3^k c + \sum_{i=1}^{k} 3^k \cdot 3^{-i} \cdot 4^i$ $\quad (a^b \cdot a^c = a^{b+c})$

$= 3^k c + \sum_{i=1}^{k} 3^k \cdot \left(\frac{4}{3}\right)^i$ $\quad (a^{-b} = \frac{1}{a^b}, a^c \cdot b^c = (ab)^c)$

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences by substituting the parameter by **4 times the parameter**.

kth substitution: $T(4^k) = 3^k c + \sum_{1}^{k} 3^k \cdot \left(\frac{4}{3}\right)^i = 3^k c + 3^k \cdot \boxed{\sum_{i=1}^{k} \left(\frac{4}{3}\right)^i}$

Summation of **the first k terms** of a geometric series with the first term being $(4/3)^1$ and the constant ratio being $4/3$.

$$= 3^k c + 3^k \cdot \left(\frac{4}{3}\right)^1 \cdot \frac{1 - \left(\frac{4}{3}\right)^k}{1 - \frac{4}{3}} \equiv \underline{3^k c + 3^k \cdot 4\left(\left(\frac{4}{3}\right)^k - 1\right)}$$

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 1

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences by substituting the parameter by **4 times the parameter**.

kth substitution: $T(4^k) = 3^k c + 3^k \cdot 4\left(\left(\frac{4}{3}\right)^k - 1\right) = 3^k c + 3^k \cdot 4 \cdot \left(\frac{4}{3}\right)^k - 3^k \cdot 4$

$$= 3^k c + 4^{k+1} - 3^k \cdot 4$$

$$= \underline{3^k(c-4) + 4^{k+1}}$$

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 2

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 2**: Let the kth previous recurrence be $T(n)$. Solve for k.

    Let $\underline{T(4^k)} = 3^k(c-4) + 4^{k+1} = T(n)$

    $\Longrightarrow n = 4^k$

    $\Longrightarrow \underline{k = \log_4 n}$

FORWARD SUBSTITUTION TO MAKE A GOOD GUESS STEP 3

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 3**: Plug $n = 4^k$ and $k = \log_4 n$ in function $T(4^k)$.

$$T(n) = \underline{3^{\log_4 n}(c-4) + 4^{\log_4 n + 1}} = 3^{\log_4 n}(c-4) + 4^{\log_4 n} \cdot 4 \qquad (a^b \cdot a^c = a^{b+c})$$

$$= \underline{n^{\log_4 3}(c-4) + n \cdot 4} \qquad (a^{\log_b c} = c^{\log_b a}, a = b^{\log_b a})$$

$$= O(n)$$

FORWARD SUBSTITUTION REVIEW

- **Step 1**: Start with **bottoms-out case**. Repeatedly derive the previous k recurrences.
  - Show at least 3 substitutions.

- **Step 2**: Let the $\boldsymbol{k}$th previous recurrence be $\boldsymbol{T(n)}$. Solve for $\boldsymbol{k}$.
  - $\boldsymbol{k}$ can be expressed as a function of $\boldsymbol{n}$.

- **Step 3**: Plug function of $\boldsymbol{k}$ (in terms of $\boldsymbol{n}$) in the $\boldsymbol{k}$th previous recurrence function. **Guess** the bound.

FORWARD SUBSTITUTION PRACTICE

- **Guess** the bound of recurrence $T(n) = \begin{cases} 1, & n = 1 \\ 2 \cdot T\left(\frac{n}{2}\right) + 1, & n > 1 \end{cases}$ using **forward** substitution.

- **Step 1**:

  1st substitution: $T(\quad) =$ ____________________

  2nd substitution: $T(\quad) =$ ____________________

  3rd substitution: $T(\quad) =$ ____________________

  …

  kth substitution: $T(\quad) =$ ____________________ $=$ ____________________

FORWARD SUBSTITUTION PRACTICE

- **Guess** the bound of recurrence $T(n) = \begin{cases} 1, & n = 1 \\ 2 \cdot T\left(\frac{n}{2}\right) + 1, & n > 1 \end{cases}$ using **forward** substitution.

- **Step 2**: Let ________ $= \boldsymbol{T(n)}$. Solve for $\boldsymbol{k}$. $\boldsymbol{k} =$ _______________
- **Step 3**:

  $T(\boldsymbol{n}) =$ ____________________

  $=$ ____________________
- **Guess** $T(n) =$

GUESS-AND-VERIFICATION

- Done with the **guess** step.
  - Empirical guess
  - Backward substitution (starts with $T(n)$ and keeps expanding $T(n)$)
  - Forward substitution (starts with the ***base case*** and keeps backtracking the recurrence to $T(n)$)

- Proceed to **verify** the **guess**.

- The **guess**-and-**verification** method is also known as the substitution method.

VERIFICATION OF THE GUESS

- Use ***mathematical induction***
  - Make the ***inductive hypothesis*** that a statement holds true for a given value n.
  - Substitute the next successive term for n
  - Prove that the statement still holds true.
  - Prove the **boundary case** holds true.
    - Note that the **boundary case** is not necessarily the **base case** of the recurrence.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP A

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step a**: Make the ***inductive hypothesis***.
    - Assume that the bound $\underline{\boldsymbol{O(n \cdot \lg n)}}$ holds for all positive $m < n$.
    - By definition, $\underline{\boldsymbol{0 \leq T(m) \leq c \cdot m \lg m}}$ for an appropriate choice of constant $c > 0$.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP B

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step b**: Pick a value for m. Substitute m in $\boldsymbol{0 \le T(m) \le c \cdot m \lg m}$.
    - In particular, let $m =$ $\underline{\boldsymbol{\lfloor n/2 \rfloor}}$, yielding $\underline{\boldsymbol{0 \le T(\lfloor n/2 \rfloor) \le c \cdot \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor}}$.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP C

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step c**: Substitute the inequality $\mathbf{0 \le T(\lfloor n/2 \rfloor) \le c \cdot \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor}$ in the original recursive function $T(n)$.

    $\mathbf{0 \le T(\lfloor n/2 \rfloor) \le c \cdot \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor} \Rightarrow \mathbf{0 \le 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \le 2c \cdot \left\lfloor \frac{n}{2} \right\rfloor \lg \lfloor \frac{n}{2} \rfloor}$ (Multiply by 2)

    $\Rightarrow \mathbf{n \le} \quad \mathbf{T(n)} \quad \mathbf{\le 2c \cdot \left\lfloor \frac{n}{2} \right\rfloor \lg \left\lfloor \frac{n}{2} \right\rfloor} + \boldsymbol{n}$ (Add $\boldsymbol{n}$)

    $\Rightarrow \underline{\mathbf{n \le T(n) \le 2c \cdot \left\lfloor \frac{n}{2} \right\rfloor \lg \left\lfloor \frac{n}{2} \right\rfloor + n}}$

    $\mathbf{0 < n \le T(n)}$ is trivially satisfied.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP D

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step d**: Derive $T(n) \le c \cdot n \lg n$ based off the inequality resulted from the substitution.

$$
\begin{aligned}
\boldsymbol{T(n)} &\le \boldsymbol{2c \cdot \left\lfloor \frac{n}{2} \right\rfloor \lg \left\lfloor \frac{n}{2} \right\rfloor + n} \\
&\le \underline{\boldsymbol{2c \cdot \left(\frac{n}{2}\right) \lg \left(\frac{n}{2}\right) + n}} \qquad (\lfloor x \rfloor \le x) \\
&= \underline{\boldsymbol{c \cdot n \lg n - cn \lg 2 + n}} \qquad \left(\log_c \frac{a}{b} = \log_c a - \log_c b\right) \\
&\le \underline{\boldsymbol{c \cdot n \lg n}} \quad \text{for } c \ \underline{\boldsymbol{\ge 1}}.
\end{aligned}
$$

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP E

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step e**: Proof the $T(n) = O(n \lg n)$ for **boundary case**.
    - Start off with the **bottoms-out case** of the recursion.
    - Assume that $\boldsymbol{T(1) = \Theta(1) = 1}$
    - Calculate $\boldsymbol{T(2), T(3)}$ based off the **bottoms-out case** and the recurrence.

$\boldsymbol{T(2) = 2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + 2 = 4}$ $\boldsymbol{T(3) = 2T\left(\left\lfloor\frac{3}{2}\right\rfloor\right) + 3 = 5}$

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP E

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step e**: Proof the $T(n) = O(n \lg n)$ for **boundary case**.
    - Let $n =$ __$\mathbf{1}$__. Plug n in $T(n) \leq c \cdot n \lg n$, yielding __$\boldsymbol{T(1) \leq c \cdot 1 \cdot \lg 1 = 0}$__.
      - Based on the **bottoms-out case** of the running time, $\boldsymbol{T(1) = \Theta(1) = 1}$
      - No choice of $c > 0$ will satisfy $\mathbf{0 = 1}$.
      - The inductive hypothesis does not hold for $n =$ __$\mathbf{1}$__.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP E

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step e**: Proof the $T(n) = O(n \lg n)$ for **boundary case**.
    - Remove $\boldsymbol{n = 1}$ from the consideration in the inductive proof.
    - Let $n =$ __**2**__. Plug n in $T(n) \le c \cdot n \lg n$, yielding __$\boldsymbol{T(2) \le c \cdot 2 \cdot \lg 2 = 2c}$__.
      - Based on the **bottoms-out case** of the running time, $\boldsymbol{T(2) = 4}$
      - Plug $\boldsymbol{T(2) = 4}$ in the inequality of the inductive hypothesis. $\boldsymbol{4 \le c \cdot 2 \cdot \lg 2 = 2c \implies c \ge 2}$
      - The inductive hypothesis holds true for $n =$ __**2**__ and c __$\boldsymbol{\ge 2}$__.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP E

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step e**: Proof the $T(n) = O(n \lg n)$ for **boundary case**.
    - The inductive hypothesis holds true for $\boldsymbol{n = 2}$ and $\boldsymbol{c \geq 2}$.
    - Let $n =$ __**3**__. Plug n in $T(n) \leq c \cdot n \lg n$, yielding _$\boldsymbol{T(3) \leq c \cdot 3 \cdot \lg 3 = 3c \lg 3}$__.
      - Based on the **bottoms-out case** of the running time, $\boldsymbol{T(3) = 5}$.
      - Plug $\boldsymbol{T(3) = 5}$ in the inequality of the inductive hypothesis. $\boldsymbol{5 \leq c \cdot 3 \cdot \lg 3 \implies c \geq 2 > \frac{5}{3 \lg 3}}$
      - The inductive hypothesis holds true for $n =$ __**3**__ and c _$\boldsymbol{\geq 2}$_.

USING SUBSTITUTION METHOD TO VERIFICATION THE GUESS STEP F

- **Verify** that recurrence $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ is $\boldsymbol{O(n \cdot \lg n)}$ using **substitution** method.

- **Solution**
  - **Step f**: Conclude that the ***inductive hypothesis*** holds true for boundary cases $\underline{\ \boldsymbol{n = 2} \text{ and } \boldsymbol{n} = 3\ }$ for appropriate choice of $c\ \underline{\geq \boldsymbol{2}\qquad}$.
  - Conclude that $T(n) = \boldsymbol{O(n \cdot \lg n)}$

USING GUESS-AND-VERIFICATION TO PROVE A BOUND REVIEW

- **Step a**: Make the ***inductive hypothesis***.
  - Assume the bound holds for any $m < n$.
- **Step b**: Pick a value for m. Substitute m in the inequality of the asymptotic definition.
- **Step c**: Substitute the inequality in the original recursive function $T(n)$.
- **Step d**: Derive the inequality of the asymptotic definition for $T(n)$.
- **Step e**: Proof the bound holds for **boundary case(s)**.
- **Step f**: Conclude that the ***inductive hypothesis*** holds true for boundary cases and for appropriate choice of c.

USING GUESS-AND-VERIFICATION TO PROVE A BOUND PRACTICE

- **Verify** that recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ is $O(n)$ using **substitution** method.

- **Step a**: Make the ***inductive hypothesis***.
  - Assume that the bound ________________ holds for all positive $m < n$.
  - By definition, ______________________ for an appropriate choice of constant $c > 0$.

USING GUESS-AND-VERIFICATION TO PROVE A BOUND PRACTICE

- **Verify** that recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ is $O(n)$ using **substitution** method.

- **Step b**: Pick a value for m. Substitute m in $\mathbf{0} \leq \boldsymbol{T}(\boldsymbol{m}) \leq \boldsymbol{c} \cdot \boldsymbol{m}$.
  - In particular, let $m =$ ________, yielding ____________________________________.

USING GUESS-AND-VERIFICATION TO PROVE A BOUND PRACTICE

- **Verify** that recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ is $O(n)$ using **substitution** method.

- **Step c**: Substitute the inequality in the original recursive function $T(n)$.

USING GUESS-AND-VERIFICATION TO PROVE A BOUND PRACTICE

- **Verify** that recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ is $O(n)$ using **substitution** method.

- **Step d**: Derive ____________________ based off the inequality resulted from the substitution.

USING GUESS-AND-VERIFICATION TO PROVE A BOUND PRACTICE

- **Verify** that recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ is $O(n)$ using **substitution** method.

- **Step e**: Proof the $T(n) = O(n)$ for **boundary case**.

USING GUESS-AND-VERIFICATION TO PROVE A BOUND PRACTICE

- **Verify** that recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ is $O(n)$ using **substitution** method.

- **Step f**: Conclude that the ***inductive hypothesis*** holds true for boundary cases ________________ for appropriate choice of c___________.

NEXT UP
RECURSION TREE

- Solving recurrence

REFERENCE

BACKWARD SUBSTITUTION ANSWER TEMPLATE

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 1**:

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

$= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$

(1st expansion: $T\left(\frac{n}{4}\right) = 3 \cdot T\left(\frac{n}{4^2}\right) + \frac{n}{4^1}$)

$=$

$\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$

(2nd expansion: $T\left(\frac{n}{4^2}\right) = 3 \cdot T\left(\frac{n}{4^3}\right) + \frac{n}{4^2}$)

BACKWARD SUBSTITUTION ANSWER TEMPLATE

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 1**:

$T(n) =$ ______________________________

(**3**rd expansion: $T\left(\frac{n}{4^3}\right) = 3 \cdot T\left(\frac{n}{4^4}\right) + \frac{n}{4^3}$)

$= \cdots$

$=$

(**k**th expansion: $T\left(\frac{n}{4^k}\right) = 3 \cdot T\left(\frac{n}{4^{k+1}}\right) + \frac{n}{4^k}$)

____

$=$

____

BACKWARD SUBSTITUTION
ANSWER TEMPLATE

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **backward** substitution.
- Solution
  - **Step 2**: kth recurrence: ____________________ $= \boldsymbol{T(1)}$. Let ___________ $= \mathbf{1}$.
  - **Step 3**: Solve for k. $\Rightarrow$ ______________.
  - **Step 4**: $T(n) =$ _____________________(fill the k-th expansion) $=$ ________________________
  - Guess $T(n) =$ _______.

FORWARD SUBSTITUTION ANSWER TEMPLATE

- **Guess** the bound of recurrence $T(n) = \begin{cases} \Theta(1), & n = 1 \\ 3 \cdot T\left(\frac{n}{4}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 1:**

    1st substitution: $T(4) =$ ________________

    2nd substitution: $T(4^2) =$ ________________

    3rd substitution: $T(4^3) =$ ________________

    kth substitution: $T(4^k) =$ ________________ $=$ ________________

FORWARD SUBSTITUTION ANSWER TEMPLATE

- **Guess** the bound of recurrence $T(n) = \begin{cases} \boldsymbol{\Theta(1)}, & \boldsymbol{n = 1} \\ \mathbf{3} \cdot T\left(\frac{\boldsymbol{n}}{\mathbf{4}}\right) + n, & n > 1 \end{cases}$ using **forward** substitution.
- Solution
  - **Step 2**: Let ______ $= \boldsymbol{T(n)}$. Solve for $\boldsymbol{k}$. $\Longrightarrow$ $\boldsymbol{k} =$ ________________
  - **Step 3**: $T(\boldsymbol{n}) =$ ________________________

    $=$ ________________________
  - **Guess** $T(n) =$