

# CSC373 Worksheet 0 Solution

July 20, 2020

1. Recurrence:  $T(n) = T(n-1) + n$

Guess:  $T(n) = \mathcal{O}(n^2)$ .

I need to show  $T(n) \leq c \cdot n^2$ .

$$T(n) \leq c(n-1)^2 + n \tag{1}$$

$$= c(n^2 - 2n + 1) + n \tag{2}$$

$$= cn^2 - c2n + c + n \tag{3}$$

$$\leq cn^2 - c2n + cn + n \tag{4}$$

$$= cn^2 - cn + n \tag{5}$$

$$\leq cn^2 - cn + cn \tag{6}$$

$$= cn^2 \tag{7}$$

## Notes:

- Substitution method
  - Solves recurrences
    - \* Recurrence characterizes the running time of divide-and-conquer algorithm
  - How it works:
    1. Make a guess for the solution
    2. Use mathematical induction to prove the guess is correct or incorrect.

## Example:

Recurrence:  $T(n) = 2T(\lfloor n/2 \rfloor) + n$

Guess:  $T(n) = \mathcal{O}(n \log n)$ ,

We need to show  $T(n) \leq cn \lg n$ .

1. Assume the bound holds for all positive  $m < n$ , in particular  $m = \lfloor n/2 \rfloor$
2. Find the upper bound of  $T(m)$

$$T(\lfloor n/2 \rfloor) \leq c \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)$$

3. Show  $T(n) = 2T(\lfloor n/2 \rfloor) + n$  leads to  $T(n) \leq cn \lg n$

$$T(n) \leq 2(c \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \quad (8)$$

$$\leq cn \lg(n/2) + n \quad (9)$$

$$= cn \lg(n) - cn \lg 2 + n \quad (10)$$

$$= cn \lg(n) - cn + n \quad (11)$$

$$\leq cn \lg(n) - cn + cn \quad (12)$$

$$\leq cn \lg(n) \quad (13)$$

4. Show that the boundary holds using mathematical induction

Doesn't have information in detail. Skipping this for now.

– Making good guess

\* Three suggestions

1. Using recursion tree
2. Through practice
3. prove loose upper and lower bounds on the recurrence and then reduce the range of uncertainty

2. Recurrence:  $T(n) = T(\lceil n/2 \rceil) + 1$

Guess:  $T(n) = \mathcal{O}(\lg n)$ .

I need to show  $T(n) \leq c \cdot \lg n$ .

$$T(n) \leq c \lg(\lceil n/2 \rceil) + 1 \quad (1)$$

$$\leq c \lg(n/2) + 1 \quad (2)$$

$$= c(\lg n - \lg 2) + 1 \quad (3)$$

$$= c(\lg n - 1) + 1 \quad (4)$$

$$= c \lg n - c + 1 \quad (5)$$

$$\leq c \lg n - c + c \quad (6)$$

**Correct Solution:**

Recurrence:  $T(n) = T(\lceil n/2 \rceil) + 1$

Guess:  $T(n) = \mathcal{O}(\lg n)$ .

I need to show  $T(n) \leq c \cdot \lg n$ .

$$T(n) \leq c \lg(\lceil n/2 \rceil) + 1 \quad (1)$$

$$\leq c \lg(n/2) + 1 \quad (2)$$

$$= c(\lg n - \lg 2) + 1 \quad (3)$$

$$= c(\lg n - 1) + 1 \quad (4)$$

$$= c \lg n - c + 1 \quad (5)$$

$$\leq c \lg n - c + c \quad (6)$$

The solution holds for  $c \geq 1$ .

3. Recurrence:  $T(n) = 2T(\lfloor n/2 \rfloor) + n$

Guess (Upperbound):  $T(n) = \mathcal{O}(n \lg n)$ .

I first need to show  $T(n) \leq c \cdot n \lg n$ .

$$T(n) = 2T(\lfloor n/2 \rfloor) + n \quad (1)$$

$$= 2c \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor + n \quad (2)$$

$$\leq 2c \cdot (n/2) \lg(n/2) + n \quad (3)$$

$$= c \cdot n(\lg n - 1) + n \quad (4)$$

$$= cn \lg n - cn + n \quad (5)$$

$$\leq cn \lg n - cn + cn \quad (6)$$

$$\leq cn \lg n \quad (7)$$

The above inequality holds for  $c \geq 1$ .

Guess (Lowerbound):  $T(n) = \Omega(n \lg n)$ .

I first need to show  $d \cdot (n - 2) \lg(n - 2) \leq T(n)$ .

$$T(n) = 2T(\lfloor (n - 2)/2 \rfloor) + n \quad (8)$$

$$\geq 2d \lfloor (n - 2)/2 \rfloor \lg \lfloor (n - 2)/2 \rfloor + n \quad (9)$$

$$\geq 2d \cdot ((n - 2)/2) \lg((n - 2)/2) + n \quad (10)$$

$$= d \cdot (n - 2)(\lg(n - 2) - 1) + n \quad (11)$$

$$= d \cdot (n - 2) \lg(n - 2) - d \cdot (n - 2) + n \quad (12)$$

$$\geq d \cdot (n - 2) \lg(n - 2) - d \cdot (n - 2) + (n - 2) \quad (13)$$

$$\geq d \cdot (n - 2) \lg(n - 2) - d \cdot (n - 2) + d \cdot (n - 2) \quad (14)$$

$$= d \cdot (n - 2) \lg(n - 2) \quad (15)$$

The above inequality holds for  $0 \leq d < 1$ .

### Notes:

- Both upper bound and lower bound don't need to be the same

#### 4.3-3

We saw that the solution of  $T(n) = 2T(\lfloor n/2 \rfloor) + n$  is  $O(n \lg n)$ . Show that the solution of this recurrence is also  $\Omega(n \lg n)$ . Conclude that the solution is  $\Theta(n \lg n)$ .

First, we guess  $T(n) \leq cn \lg n$ , ← upper bound

$$\begin{aligned} T(n) &\leq 2c \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor + n \\ &\leq cn \lg(n/2) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n + (1 - c)n \\ &\leq cn \lg n, \end{aligned}$$

where the last step holds for  $c \geq 1$ .

Next, we guess  $T(n) \geq c(n + 2) \lg(n + 2)$ , ← lower bound

$$\begin{aligned} T(n) &\geq 2c(\lfloor n/2 \rfloor + 2)(\lg(\lfloor n/2 \rfloor + 2) + 1) + n \\ &\geq 2c(n/2 - 1 + 2)(\lg(n/2 - 1 + 2) + 1) + n \\ &= 2c \frac{n+2}{2} \lg \frac{n+2}{2} + n \\ &= c(n+2) \lg(n+2) - c(n+2) \lg 2 + n \\ &= c(n+2) \lg(n+2) + (1 - c)n - 2c \\ &\geq c(n+2) \lg(n+2), \end{aligned}$$

where the last step holds for  $n \geq \frac{2c}{1-c}$ ,  $0 \leq c < 1$ .

#### 4. Recurrence (Merge sort):

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{if } n > 1 \end{cases}$$

Guess (upper bound):  $T(n) \leq c \cdot (n - 2) \cdot \lg(n - 2)$

$$T(n) \leq c(\lceil n/2 \rceil - 2) \lg(\lceil n/2 \rceil - 2) + c(\lfloor n/2 \rfloor - 2) \lg(\lfloor n/2 \rfloor - 2) + dn \quad (1)$$

$$= c(n/2 + 1 - 2) \lg(n/2 + 1 - 2) + c(n/2 + 1 - 2) \lg(n/2 + 1 - 2) + dn \quad (2)$$

$$= c((n - 2)/2) \lg((n - 2)/2) + c((n - 2)/2) \lg((n - 2)/2) + dn \quad (3)$$

$$= c(n - 2) \lg((n - 2)/2) + dn \quad (4)$$

$$= c(n - 2) \lg(n - 2) - c(n - 2) + dn \quad (5)$$

$$= c(n - 2) \lg(n - 2) - (d - c)n + 2c \quad (6)$$

$$= c(n - 2) \lg(n - 2) \quad (7)$$

The bound holds as long as  $c > d$ .

Guess (lower bound):  $c \cdot (n - 2) \cdot \lg(n - 2) \leq T(n)$

$$T(n) \leq c(\lceil n/2 \rceil + 1) \lg(\lceil n/2 \rceil + 1) + c(\lfloor n/2 \rfloor + 1) \lg(\lfloor n/2 \rfloor + 1) + dn \quad (8)$$

$$\leq c(n/2 - 1 + 1) \lg(n/2 - 1 + 1) + c(n/2 - 1 + 1) \lg(n/2 - 1 + 1) + dn \quad (9)$$

$$= c(n/2) \lg(n/2) + c(n/2) \lg(n/2) + dn \quad (10)$$

$$= cn \lg(n/2) + dn \quad (11)$$

$$= cn \lg(n) - cn + dn \quad (12)$$

$$= cn \lg(n) + (d - c)n \quad (13)$$

$$\leq c(n - 1) \lg(n - 1) \quad (14)$$

The bound holds as long as  $d > c$ , and  $0 \leq c < 1$

### Notes:

- the  $n$  here is asymptotically large

5. Recurrence:  $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$

Guess (upper bound):  $cn \lg n$

$$T(n) \leq 2c(\lfloor n/2 \rfloor + 17) \lg(\lfloor n/2 \rfloor + 17) + n \quad (15)$$

$$\leq 2c((n/2) + 17) \lg((n/2) + 17) + n \quad (16)$$

$$= 2c(n/2) \lg(n/2) + n \quad (17)$$

$$= cn(\lg(n) - 1) + n \quad (18)$$

$$= cn \lg(n) - cn + n \quad (19)$$

$$\leq cn \lg(n) - cn + cn \quad (20)$$

$$= cn \lg(n) \quad (21)$$

6.

$$T(n) = 4T(n/3) + n \quad (1)$$

$$\leq 4c(n/3)^{\log_3 4} + n \quad (2)$$

$$\leq 4c(1/3)^{\log_3 4} n^{\log_3 4} + n \quad (3)$$

$$\leq (4/4)cn^{\log_3 4} + n \quad (4)$$

$$\leq cn^{\log_3 4} + n \quad (5)$$

We cannot advance further since  $n$  in  $cn^{\log_3 4} + n$  cannot be eliminated.

With the new guess  $T(n) \leq cn^{\log_3 4} - dn$ , we have

$$T(n) = 4T(n/3) + n \quad (6)$$

$$\leq 4c(n/3)^{\log_3 4} - d(n/3) + n \quad (7)$$

$$= 4c(n/3)^{\log_3 4} - d(n/3) + n \quad (8)$$

$$= (4/3^{\log_3 4})cn^{\log_3 4} - d(n/3) + n \quad (9)$$

$$= (4/4)cn^{\log_3 4} - d(n/3) + n \quad (10)$$

$$= cn^{\log_3 4} - d(n/3) + n \quad (11)$$

$$\leq cn^{\log_3 4} - d(n/3) + n \quad (12)$$

$$\leq cn^{\log_3 4} \quad (13)$$

The bound holds as long as  $d \geq 3$  and  $c \geq 1$ .

**Correct Solution:**

Recurrence:  $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$

Guess (upper bound):  $cn \lg n$

$$T(n) \leq 2c(\lfloor n/2 \rfloor + 17) \lg(\lfloor n/2 \rfloor + 17) + n \quad (14)$$

$$\leq 2c((n/2) + 17) \lg((n/2) + 17) + n \quad (15)$$

$$= 2c(n/2) \lg(n/2) + n \quad (16)$$

$$= cn(\lg(n) - 1) + n \quad (17)$$

$$= cn \lg(n) - cn + n \quad (18)$$

$$\leq cn \lg(n) - cn + cn \quad (19)$$

$$= cn \lg(n) \quad (20)$$

$$T(n) = 4T(n/3) + n \quad (1)$$

$$\leq 4c(n/3)^{\log_3 4} + n \quad (2)$$

$$\leq 4c(1/3)^{\log_3 4} n^{\log_3 4} + n \quad (3)$$

$$\leq (4/4)cn^{\log_3 4} + n \quad (4)$$

$$\leq cn^{\log_3 4} + n \quad (5)$$

We cannot advance further since  $n$  in  $cn^{\log_3 4} + n$  cannot be eliminated.

With the new guess  $T(n) \leq cn^{\log_3 4} - dn$ , we have

$$T(n) = 4T(n/3) + n \quad (6)$$

$$\leq 4c(n/3)^{\log_3 4} - 4d(n/3)4d(n/3) + n \quad (7)$$

$$= 4d(n/3) = 4c(n/3)^{\log_3 4} - 4d(n/3) + n \quad (8)$$

$$= 4d(n/3) = (4/3^{\log_3 4})cn^{\log_3 4} - 4d(n/3) + n \quad (9)$$

$$= (4/4)cn^{\log_3 4} - 4d(n/3) + n \quad (10)$$

$$= cn^{\log_3 4} - 4d(n/3) + n \quad (11)$$

$$\leq cn^{\log_3 4} - 4d(n/3) + n \quad (12)$$

$$\leq cn^{\log_3 4} - 4d(n/2) + n \quad (13)$$

$$\leq cn^{\log_3 4} - 2dn + n \quad (14)$$

$$\leq cn^{\log_3 4} - 2dn + dn \quad (15)$$

$$\leq cn^{\log_3 4} - dn \quad (16)$$

7. I need to show  $T(n) \leq cn^2$

$$T(n) = 4T(n/2) + n \quad (17)$$

$$\leq 4c(n/2)^2 + n \quad (18)$$

$$= (4/4)cn^2 + n \quad (19)$$

$$= cn^2 + n \quad (20)$$

We cannot advance further since  $n$  in  $cn^2 + n$  cannot be eliminated.

But with the new guess  $T(n) \leq cn^2 - dn$ , we have

$$T(n) = 4T(n/2) + n \quad (21)$$

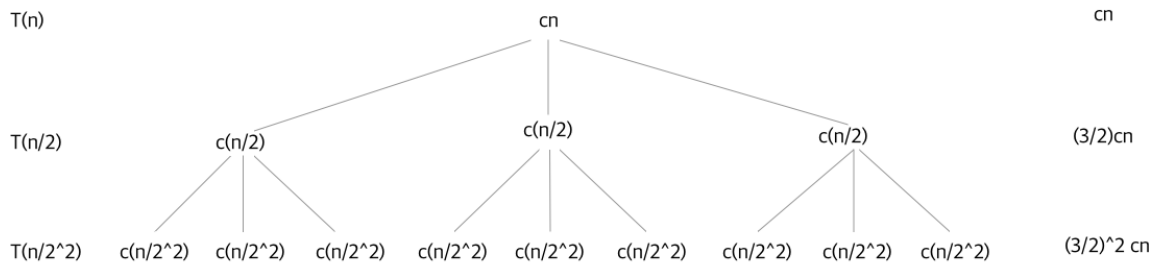
$$\leq 4c(n/2)^2 - 4d(n/2) + n \quad (22)$$

$$= (4/4)cn^2 - 2dn + n \quad (23)$$

$$\leq cn^2 - 2dn + dn \quad (24)$$

$$= cn^2 - dn \quad (25)$$

The bound holds as long as  $d \geq 1$  and  $c \geq 1$ .



8.

1. Finding number of levels in recursion tree

$$1 = n/2^i \quad (1)$$

$$2^i = n \quad (2)$$

$$i = \log_2 n \quad (3)$$



## 2. Finding the total cost of recursion tree

The tree has  $n^{\lg 3}$  leaves. So, we have

$$T(n) = n \cdot \sum_{i=0}^{\lg_2(n)-1} (3/2)^i + \Theta(n^{\lg 3}) \quad (4)$$

$$= n \cdot \left( \frac{(3/2)^{\lg_2(n)} - 1}{(3/2) - 1} \right) + \Theta(n^{\lg 3}) \quad (5)$$

$$= 2n \cdot \left( (3/2)^{\lg_2(n)} - 1 \right) + \Theta(n^{\lg 3}) \quad (6)$$

$$= 2n \cdot \left( n^{\lg(3/2)} - 1 \right) + \Theta(n^{\lg 3}) \quad (7)$$

$$= 2n \cdot \left( n^{\lg(3/2)} - 1 \right) + \Theta(n^{\lg 3}) \quad (8)$$

$$= 2 \cdot \left( n^{\lg 3 - 1 + 1} - n \right) + \Theta(n^{\lg 3}) \quad (9)$$

$$= 2 \cdot \left( n^{\lg 3} - n \right) + \Theta(n^{\lg 3}) \quad (10)$$

$$= 2 \cdot \left( n^{\lg 3} - n \right) + \Theta(n^{\lg 3}) \quad (11)$$

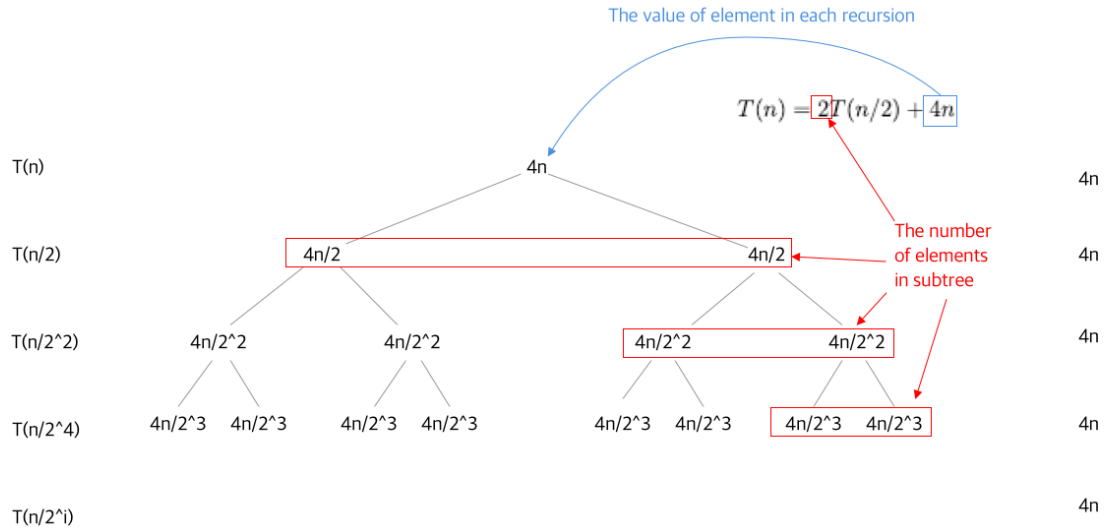
Thus, the guess for the upper bound is  $T(n) = \mathcal{O}(n^{\lg 3})$

**Notes:**

- Recursion Tree
  - Provides a straightforward way to provide a good guess.
  - Is then verified using substitution method

**Example:**

Recurrence:  $T(n) = 2T(n/2) + 4n, T(1) = 4$



1. Finding number of levels in recursion tree

$$1 = n/2^i \quad (12)$$

$$2^i = n \quad (13)$$

$$i = \log_2 n \quad (14)$$

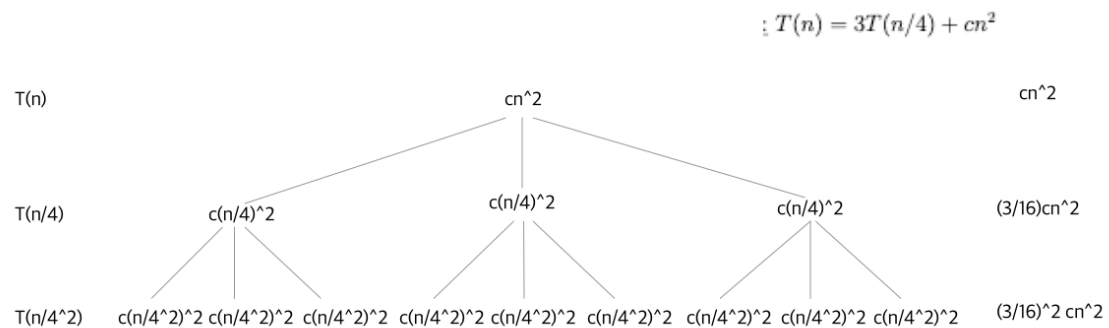
2. Finding the value of guess

$$\sum_{i=0}^{\log_2 n} 4n = 4n \cdot \sum_{i=0}^{\log_2 n} 1 \quad (15)$$

$$= 4n(\log_2 n + 1) \quad (16)$$

### Example 2:

Recurrence:  $T(n) = 3T(n/4) + cn^2$



Steps:

1. Finding number of levels in recursion tree

$$1 = n/4^i \quad (17)$$

$$4^i = n \quad (18)$$

$$i = \log_4 n \quad (19)$$

2. Finding the cost of entire tree

$$T(n) = \sum_{i=0}^{\log_4 n - 1} c(3/16)^i n^2 + \Theta(n^{\log_4 3}) \quad (20)$$

$$= cn^2 \cdot \sum_{i=0}^{\log_4 n - 1} (3/16)^i + \Theta(n^{\log_4 3}) \quad (21)$$

$$< cn^2 \cdot \sum_{i=0}^{\infty} (3/16)^i + \Theta(n^{\log_4 3}) \quad [\text{since } n \text{ is asympt. large}] \quad (22)$$

$$= cn^2 \left( \frac{1}{1 - (3/16)} \right) + \Theta(n^{\log_4 3}) \quad [\text{Since } \sum_{i=0}^{\infty} ar^i = \frac{a}{1-r}] \quad (23)$$

– **Note:**  $(\log_4(n - 1))$  because in  $i = 0, \dots, i = \log_4(n - 1)$  there are  $\log_4(n)$  elements

3. Finding the upper bound of  $T(n)$

Since the total cost is  $T(n) = cn^2 \left( \frac{1}{1 - (3/16)} \right) + \Theta(n^{\log_4 3})$ , we have  $\mathcal{O}(n^2)$

4. Verify the correctness of guess using substitution method

$$T(n) \leq 3T(\lfloor n/4 \rfloor) + cn^2 \quad (24)$$

$$\leq 3d\lfloor n/4 \rfloor^2 + cn^2 \quad (25)$$

$$\leq 3d(n/4)^2 + cn^2 \quad (26)$$

$$= (3/16)dn^2 + cn^2 \quad (27)$$

$$\leq dn^2 \quad (28)$$

where the last step holds as long as  $d \geq (16/13)c$ .