

CSC373 Worksheet 0 Solution

July 22, 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. Solution:



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 $T(n) = \mathcal{O}(n^{\lg 3})$

3. Verifying the correct guess using the substitution method

Guess: $T(n) \leq cn^{\lg 3} - dn$

I need to show the guess holds in the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$.

Indeed we have

$$T(n) = 3T(\lfloor n/2 \rfloor) + n \quad (12)$$

$$\leq 3(cn^{\lg 3} - d\lfloor n/2 \rfloor) + n \quad (13)$$

$$= 3\left(\frac{cn^{\lg 3}}{3} - d\left(\frac{n}{2} + 1\right)\right) + n \quad (14)$$

$$= 3\left(\frac{cn^{\lg 3}}{3} - \frac{3dn}{2}\right) + n \quad (15)$$

$$\leq 3\left(\frac{cn^{\lg 3}}{3} - \frac{3dn}{3}\right) + n \quad (16)$$

$$= cn^{\lg 3} - 3dn + n \quad (17)$$

$$\leq cn^{\lg 3} - 3dn + 2dn \quad (18)$$

$$= cn^{\lg 3} - dn \quad (19)$$

And the boundary holds as long as $c \geq 0$ and $d \geq 1$.

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

$$1 = n/2^i \quad (20)$$

$$2^i = n \quad (21)$$

$$i = \log_2 n \quad (22)$$

2. Finding the value of guess

$$\sum_{i=0}^{\log_2 n} 4n = 4n \cdot \sum_{i=0}^{\log_2 n} 1 \quad (23)$$

$$= 4n(\log_2 n + 1) \quad (24)$$

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

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

Steps:

1. Finding number of levels in recursion tree

$$1 = n/4^i \quad (25)$$

$$4^i = n \quad (26)$$

$$i = \log_4 n \quad (27)$$

2. Finding the cost of entire tree

$$T(n) = \sum_{i=0}^{\log_4 n - 1} c(3/16)^i n^2 + \Theta(n^{\log_4 3}) \quad (28)$$

$$= cn^2 \cdot \sum_{i=0}^{\log_4 n - 1} (3/16)^i + \Theta(n^{\log_4 3}) \quad (29)$$

$$< cn^2 \cdot \sum_{i=0}^{\infty} (3/16)^i + \Theta(n^{\log_4 3}) \quad [\text{since } n \text{ is asympt. large}] \quad (30)$$

$$= cn^2 \left(\frac{1}{1 - (3/16)} \right) + \Theta(n^{\log_4 3}) \quad \left[\text{Since } \sum_{i=0}^{\infty} ar^i = \frac{a}{1-r} \right] \quad (31)$$

– **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

$$T(n) \leq 3T(\lfloor n/4 \rfloor) + cn^2 \quad (32)$$

$$\leq 3d\lfloor n/4 \rfloor^2 + cn^2 \quad (33)$$

$$\leq 3d(n/4)^2 + cn^2 \quad (34)$$

$$= (3/16)dn^2 + cn^2 \quad (35)$$

$$\leq dn^2 \quad (36)$$

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

9. Solution:

$T(n)$	n^2	n^2
$T(n/2)$	$(n/2)^2$	$n^2/2^2$
$T(n/2^2)$	$(n/4)^2$	$n^2/2^4$
$T(n/2^3)$	$(n/8)^2$	$n^2/2^6$

1. Finding number of levels in recursion tree

$$1 = \frac{n}{2^i} \quad (1)$$

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

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

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

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

$$= n^2 \cdot \sum_{i=0}^{\infty} \frac{1}{2^{2i}} + \Theta(1) \quad [\text{since } n \text{ is asympt. large}] \quad (5)$$

$$= n^2 \cdot \left(\frac{1}{1 - \frac{1}{4}} \right) + \Theta(1) \quad (6)$$

$$= \frac{4n^2}{3} + \Theta(1) \quad (7)$$

Thus, we we can conclude $T(n) = \mathcal{O}(n^2)$

3. Verify the correctness of guess using substitution method

Guess: $T(n) \leq cn^2$

I need to show the guess holds for the recurrence $T(n) = T(\frac{n}{2}) + n$.

And, indeed we have

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

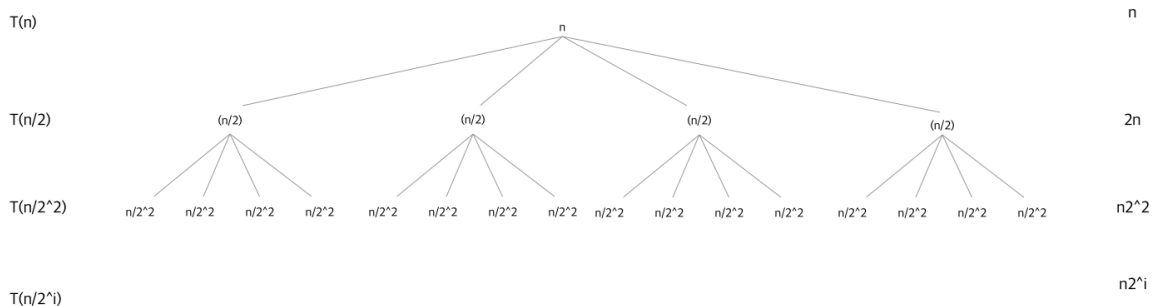
$$\leq \frac{cn^2}{4} + n^2 \quad (9)$$

$$= \left(\frac{c}{4} + 1\right) \cdot n^2 \quad (10)$$

$$\leq cn^2 \quad (11)$$

The boundary holds when $c \geq \frac{4}{3}$.

10. **Solution:**





- Find the total cost of the recursion tree

$$1 = \frac{n}{2^i} \quad (1)$$

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

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

- Finding the upper bound of $T(n)$

$$T(n) = \sum_{i=0}^{\lg n - 1} (2 \cdot 4^i + n2^i) + \Theta(n^2) \quad (4)$$

$$= \sum_{i=0}^{\lg n - 1} 2 \cdot 4^i + \sum_{i=0}^{\lg n - 1} n2^i + \Theta(n^2) \quad (5)$$

$$= 2 \cdot \sum_{i=0}^{\lg n - 1} 4^i + n \cdot \sum_{i=0}^{\lg n - 1} 2^i + \Theta(n^2) \quad (6)$$

$$= 2 \cdot \left(\frac{4^{\lg n} - 1}{4 - 1} \right) + n \cdot (n - 1) + \Theta(n^2) \quad \left[\text{Since } \sum_{i=0}^{n-1} ar^i = a \cdot \frac{r^n - 1}{r - 1}, \text{ where } r \neq 1 \right] \quad (7)$$

$$= \frac{2}{3} \cdot (n^2 - 1) + n \cdot (n - 1) + \Theta(n^2) \quad (8)$$

$$= \mathcal{O}(n^2) + \Theta(n^2) \quad (9)$$

$$(10)$$

- Verify the correctness of guess using substitution method

Guess: $T(n) \leq cn^2 - dn$.

I need to show the guess holds for the recurrence $T(n) = 4T(\frac{n}{2} + 2) + n$

$$T(n) = 4T\left(\frac{n}{2} + 2\right) + n \quad (11)$$

$$\leq 4c\left(\frac{n}{2} + 2\right)^2 - 4dn + n \quad (12)$$

$$= 4c\left(\frac{n^2}{4} + 2n + 4\right) - 4dn + n \quad (13)$$

$$\leq cn^2 - 4dn + n \quad [\text{Since } n^2 \text{ dominates } n \text{ asymptotically}] \quad (14)$$

$$\leq cn^2 - 4dn + 3dn \quad (15)$$

$$= cn^2 - dn \quad (16)$$

Correct Solution:



- Find the total cost of the recursion tree

$$1 = \frac{n}{2^i} \quad (17)$$

$$2^i = n \quad (18)$$

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

- Finding the upper bound of $T(n)$

$$T(n) = \sum_{i=0}^{\lg n - 1} (2 \cdot 4^i + n2^i) + \Theta(n^2) \quad (20)$$

$$= \sum_{i=0}^{\lg n - 1} 2 \cdot 4^i + \sum_{i=0}^{\lg n - 1} n2^i + \Theta(n^2) \quad (21)$$

$$= 2 \cdot \sum_{i=0}^{\lg n - 1} 4^i + n \cdot \sum_{i=0}^{\lg n - 1} 2^i + \Theta(n^2) \quad (22)$$

$$= 2 \cdot \left(\frac{4^{\lg n} - 1}{4 - 1} \right) + n \cdot (n - 1) + \Theta(n^2) \quad \left[\text{Since } \sum_{i=0}^{n-1} ar^i = a \cdot \frac{r^n - 1}{r - 1}, \text{ where } r \neq 1 \right] \quad (23)$$

$$= \frac{2}{3} \cdot (n^2 - 1) + n \cdot (n - 1) + \Theta(n^2) \quad (24)$$

$$= \Theta(n^2) \quad (25)$$

- Verify the correctness of guess using substitution method

Guess: $T(n) \leq cn^2 - dn$.

I need to show the guess holds for the recurrence $T(n) = 4T(\frac{n}{2} + 2) + n$

$$T(n) = 4T(\frac{n}{2} + 2) + n \quad (26)$$

$$\leq 4c(\frac{n}{2} + 2)^2 - 4dn + n \quad (27)$$

$$= 4c(\frac{n^2}{4} + 2n + 4) - 4dn + n \quad (28)$$

$$\leq cn^2 - 4dn + n \quad \left[\text{Since } n^2 \text{ dominates } n \text{ asymptotically} \right] \quad (29)$$

$$\leq cn^2 - 4dn + 3dn \quad (30)$$

$$= cn^2 - dn \quad (31)$$

Notes:

- The solution has $4^{\lg n} = n^2$. I noticed the same for $3^{\lg n} = n^3$. I had trouble looking for relevant formulas. Is this true in general? Can I replace variables in powers with the base?
- Noticed that in solution, the total cost is found for each term in $T(\frac{n}{2} + 2)$ (i.e. first for $\frac{n}{2}$ and second for 2). and then combined together in the end.

11. Solution:

- Finding the depth of tree

$$n - 1 \quad (1)$$

- Finding the number of leaves in the tree

$$\text{number of branchings}^{\text{depth of tree}} = 2^{n-1} \quad (2)$$

- Finding the upper bound of $T(n)$

$$T(n) \leq \sum_{i=0}^{n-1} 2^i + \Theta(2^n) \quad (3)$$

$$= \left(\frac{2^n - 1}{2 - 1} \right) + \Theta(2^n) \quad (4)$$

$$= (2^n - 1) + \Theta(2^n) \quad (5)$$

$$= \Theta(2^n) \quad (6)$$

- Verify the correctness of guess using substitution method

Guess: $T(n) \geq c2^n$

I need to show the bound holds for $T(n) = 2T(n-1) + 1$.

Indeed we have

$$T(n) = 2T(n-1) + 1 \quad (7)$$

$$< 2c2^{n-1} + 1 \quad (8)$$

$$= c2^n + 1 \quad (9)$$

$$= c2^n \quad [\text{Since } n \text{ is asympt. large}] \quad (10)$$

And the boundary holds when $c \geq 1$.

Notes:

- If constant term in T exists, but The term after $T()$ is constant, then it's ignored. It is considered when it's in terms of n .
- Calculating the number of leaves

$$\text{number of branchings}^{\text{depth of tree}} \quad (11)$$

Example:

2^{n-1} (in above example)

12. Solution:

I will solve only the upper bound for now.



1. Find the depth of longest simple path in recursion tree

The longest simple path is created by $T(n-1)$ and has depth of 2^{n-1} .

2. Find the number of leaves expecting a full binary tree of the same depth

Here, the number of leaves is at most:

$$\text{number of branchings}^{\text{depth of tree}} = 2^{2^{n-1}} \quad (1)$$

3. Find the upper bound of $T(n)$ that produces most depth

$$\sum_{i=0}^{n-1} 2^i + \dots = \left(\frac{2^n - 1}{2 - 1} \right) + \dots \quad (2)$$

$$= (2^n - 1) + \dots \quad (3)$$

$$= \mathcal{O}(2^n) \quad (4)$$

4. Valiate the upper bound using the substitution method

Guess: $T(n) \leq c2^n - 2dn$

I need to show the guess holds for the recurrence $T(n) = T(n-1) + T(\frac{n}{2}) + n$.

And indeed we have

$$T(n) = T(n-1) + T(\frac{n}{2}) + n \quad (5)$$

$$\leq c2^{n-1} - 2d(n-1) + c2^{\frac{n}{2}} - 2\left(\frac{dn}{2}\right) + n \quad (6)$$

$$= c2^{n-1} - 2d(n-1) + c2^{\frac{n}{2}} - dn + n \quad (7)$$

$$\leq c2^{n-1} - 2d(n-1) + c2^{\frac{n}{2}} - dn + dn \quad (8)$$

$$= c2^{n-1} - 2d(n-1) + c2^{\frac{n}{2}} \quad (9)$$

$$= c2^{n-1} - 2d(n-1) \quad [\text{Since } c2^{n-1} \text{ dominates } c2^{\frac{n}{2}}] \quad (10)$$

$$= c2^n - 2dn \quad [\text{Since } n \text{ dominates } -1] \quad (11)$$

$$\leq c2^n - dn \quad (12)$$

And the bound holds when $c \geq 1$ (not too sure) and $d \geq 1$.

Notes:

- Solving recurrence with uneven recursion tree

Example: $T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + \mathcal{O}(n)$

1. Find the depth of longest simple path in recursion tree

The longest simple path is created in $T(\frac{2n}{3})$. With the depth of $i = \log_{3/2} n$.

2. Find the number of leaves expecting a full binary tree of the same depth

Here, the number of leaves is $\mathcal{O}(n)$.

3. Find the upper bound of $T(\dots)$ that produces most depth

$$\mathcal{O}(\text{cost at depth} \times \text{depth}) = \mathcal{O}(cn \log_{3/2} n) = \mathcal{O}(n \lg n)$$

$$- \mathcal{O}(cn \log_{3/2} n) \rightarrow \mathcal{O}(n \lg n) \text{ since } \frac{3}{2} < 2 \text{ (There seems to be a lot of sloppiness)}$$

4. Valiate the upper bound using the substitution method

$$T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + cn \quad (13)$$

$$\leq d(\frac{n}{3}) \cdot \lg(\frac{n}{3}) + d(\frac{2n}{3}) \lg(\frac{2n}{3}) + cn \quad (14)$$

$$= (d(\frac{n}{3}) \lg n - d(\frac{n}{3}) \cdot \lg 3) + (d(\frac{2n}{3}) \lg n - d(\frac{2n}{3}) \lg(\frac{3}{2})) + cn \quad (15)$$

$$= dn \lg n - d((\frac{n}{3} \lg 3) + (\frac{2n}{3}) \lg(3/2)) + cn \quad (16)$$

$$= dn \lg n - d((\frac{n}{3}) \lg 3 + (\frac{2n}{3}) \lg(3) - (\frac{2n}{3}) \lg(2)) + cn \quad (17)$$

$$= dn \lg n - dn(\lg 3 - \frac{2}{3}) + cn \quad (18)$$

$$\leq dn \lg n \quad (19)$$

And the above is true as long as $d \geq \frac{c}{\lg 3 - \frac{2}{3}}$

- I don't feel too sure about how to calculate the number of leaf nodes.

13. The shortest simple path from the root occurs in $T(\frac{n}{3})$ with the value of $i = \log_3 n$.

The figure 4.6 tells us each level in the recurrence tree has cost of cn .

Since the solution to the recurrence is at least the number of levels times the cost of each level, the solution is $\Omega(cn \log_3 n) = \Omega(\frac{cn \lg n}{\lg 3}) = \Omega(n \lg n)$.

14. **Solution:**

1. Finding the depth of tree

The longest simple path from the root to a leaf is $n \rightarrow \frac{n}{2} \rightarrow \frac{n}{4} \rightarrow \dots \rightarrow 1$.

Since $(\frac{n}{2^i} = 1)$ when $i = \lg n$, the height of the tree is $\lg n$.

2. Finding the cost at each level in the tree

Each level has four times more nodes than the level above.

So, the number of nodes at depth i is 4^i .

Now, each node at $i = 0, \dots, \lg(n) - 1$ has cost of $\frac{cn}{2^i}$.

So, by multiplying together, the cost of all nodes at depth i is $cn2^i$

3. Finding the cost of leaf nodes

The bottom level, at depth $\lg n$ has $n^{\lg 4} = n^2$ nodes with the cost of $n^2 T(1)$ or $\Theta(n^2)$.

4. Finding the total cost of $T(n)$, or the tight asymptotic bound

$$T(n) \leq \sum_{i=0}^{\lg n - 1} cn2^i + \Theta(n^2) \quad (1)$$

$$= cn \left(\frac{2^{\lg n} - 1}{2 - 1} \right) + \Theta(n^2) \quad (2)$$

$$= cn(n - 1) + \Theta(n^2) \quad (3)$$

$$= \Theta(n^2) \quad (4)$$

Thus, the tight asymptotic bound is $\Theta(n^2)$.

5. Verifying the upper bound

Let the guess be $T(n) \leq dn^2 - en$.

I need to show the guess holds for the recurrence $T(n) = 4T(\lfloor n/2 \rfloor) + cn$.

Indeed we have

$$T(n) = 4T(\lfloor n/2 \rfloor) + cn \tag{5}$$

$$\leq 4d\lfloor \frac{n}{2} \rfloor^2 - 4e\lfloor \frac{n}{2} \rfloor + cn \tag{6}$$

$$\leq 4d\left(\frac{n}{2}\right)^2 - 4e\left(\frac{n}{2} - 1\right) + cn \tag{7}$$

$$= 4d\left(\frac{n^2}{4}\right) - e(2n - 4) + cn \tag{8}$$

$$= dn^2 - e(2n - 4) + cn \tag{9}$$

$$= dn^2 - e(2n - 4) + cn \quad [\text{since } n \text{ is asympt. large}] \tag{10}$$

$$= dn^2 - e2n + cn \tag{11}$$

$$= dn^2 - n(e2 - c) \tag{12}$$

$$\leq dn^2 - ne \tag{13}$$

as long as $c \geq e$ and $d \geq 1$.

6. Verifying the lower bound

Let the guess be $d(n + 2)^2 \leq T(n)$.

I need to show the guess holds for the recurrence $T(n) = 4T(\lfloor n/2 \rfloor) + cn$.

Indeed we have

$$T(n) = 4T(\lfloor n/2 \rfloor) + cn \quad (14)$$

$$\geq 4d(\lfloor \frac{n}{2} \rfloor + 2)^2 + cn \quad (15)$$

$$\geq 4d(\frac{n}{2} - 1 + 2)^2 + cn \quad (16)$$

$$= 4d\left(\frac{n}{2} + 1\right)^2 + cn \quad (17)$$

$$= d(n + 2)^2 + cn \quad (18)$$

$$\geq d(n + 2)^2 \quad (19)$$

as long as $c \geq 0$ and $d \geq 1$.

15. a) Here we have $a = 2, b = 4, f(n) = 1$. Since $1 = n^0 = n^{\log_4(2) - \log_4(2)}$ where $\epsilon = \log_4(2)$, the case 1 of master's theorem applies, and $T(n) = \Theta(n^{\log_4 2})$

Notes:

- Master method
 - The master method provides a "cookbook" method for solving recurrences of the form

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \quad (1)$$

where $a \geq 1$ and $b > 1$ are constants and $f(n)$ is asymptotically positive function.

- Allows to solve problems without pencil or paper
- Master Theorem
 - Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

where we interpret $\frac{n}{b}$ to be either $\lfloor \frac{n}{b} \rfloor$ or $\lceil \frac{n}{b} \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. if $f(n) = \mathcal{O}(n^{\log_b(a-\epsilon)})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. if $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$
3. if $f(n) = \Omega(n^{\log_b(a+\epsilon)})$ for some constant $\epsilon > 0$ and if $af(\frac{n}{b}) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n , then $T(n) = \Theta(f(n))$.

- Comparing $f(n)$ with the function $n^{\log_b a}$, the larger of two functions determine the solution to the recurrence.

Example:

$$T(n) = 9T\left(\frac{n}{3}\right) + n$$

Here $a = 9, b = 3, f(n) = n$. Since $f(n) = \mathcal{O}(n^{\log_3 9 - \epsilon})$ where $\epsilon = 1$, the case 1 of master theorem tells us $T(n) = \Theta(n^2)$.

Example #2:

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

Here, $a = 1, b = \frac{3}{2}, f(n) = 1$. Since $f(n) = 1 = n^0 = n^{\log_{3/2} 1}$, the case 2 of master theorem applies and $T(n) = \Theta(\lg n)$.

- b) Here $a = 2, b = 4$ and $f(n) = n^{\frac{1}{2}}$. Since $f(n) = n^{\frac{1}{2}} = n^{\log_4 2}$, the case 2 of master theorem applies and $T(n) = \Theta(n^{\log_4 2}) = \Theta(\sqrt{n})$.

Correct Solution:

Here $a = 2, b = 4$ and $f(n) = n^{\frac{1}{2}}$. Since $f(n) = n^{\frac{1}{2}} = n^{\log_4 2}$, the case 2 of master theorem applies and $T(n) = \Theta(n^{\log_4 2}) = \Theta(\sqrt{n} \lg n)$.

- c) Here, we have $a = 2, b = 4, f(n) = n$ and $n^{\log_4 2} = \mathcal{O}(n^{0.5})$.

Since $f(n) = \Omega(n^{\log_4 2 + \epsilon})$, where $\epsilon = 0.5$, and $af\left(\frac{n}{b}\right) = 2f\left(\frac{n}{4}\right) = \frac{n}{2} \leq cn$ where $c = \frac{1}{2}$ for sufficiently large n , the case 3 of master theorem applies.

Thus, we have $T(n) = \Theta(n)$.

- d) Here, we have $a = 2, b = 4, f(n) = n^2$ and $n^{\log_4 2} = \mathcal{O}(n^{0.5})$.

Since $f(n) = \Omega(n^{\log_4 2 + \epsilon})$, where $\epsilon = 1.5$, and $af\left(\frac{n}{b}\right) = 2f\left(\frac{n}{4}\right) = \frac{n^2}{8} \leq cn$ where $c = \frac{1}{8}$ for sufficiently large n , the case 3 of master theorem applies.

Thus, we have $T(n) = \Theta(n^2)$.

16. Here $a = 1, b = 2$, and $f(n) = 1$.

Since $f(n) = 1 = n^0 = n^{\log_2 1} = n^{\log_b a}$, case 2 of master's theorem applies.

Thus, we have $T(n) = \Theta(n^{\log_2 1} \lg n) = \Theta(\lg n)$.

17. $f(n) = n \lg n$ cannot be represented in the form $n^{\log_b a + \epsilon}$.

Thus, master's method cannot be applied here.

So, to determine asymptotic upper bound, recurrence tree method + substitution method is used.

Using the recurrence tree method, the recurrence tree has depth of $\lg n$, the level cost of $2^i n^2 \lg n$ where $i = 0, 1, \dots, \lg n - 1$, and the leaf cost of $\Theta(n^2)$.

So, the asymptotic upper bound is:

$$T(n) = +\Theta(n^2) \quad (2)$$