CSC373 Worksheet 0 Solution

July 21, 2020

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

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

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

$$
\begin{aligned}
T(n) &\leq c(n-1)^2 + n && (1)\\
&= c(n^2 - 2n + 1) + n && (2)\\
&= cn^2 - c2n + c + n && (3)\\
&\leq cn^2 - c2n + cn + n && (4)\\
&= cn^2 - cn + n && (5)\\
&\leq cn^2 - cn + cn && (6)\\
&= cn^2 && (7)
\end{aligned}
$$

Notes:

- Substitution method
  - Solves recurrences
    - Recurrence characters 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 $T(m)$

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

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

$$
\begin{align}
T(n) &\le 2(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \tag{8}\\
&\le cn \lg(n/2) + n \tag{9}\\
&= cn \lg(n) - cn \lg 2 + n \tag{10}\\
&= cn \lg(n) - cn + n \tag{11}\\
&\le cn \lg(n) - cn + cn \tag{12}\\
&\le cn \lg(n) \tag{13}
\end{align}
$$

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 $T(n) \le c \cdot \lg n$.

$$
\begin{align}
T(n) &\le c \lg(\lceil n/2 \rceil) + 1 \tag{1}\\
&\le c \lg(n/2) + 1 \tag{2}\\
&= c(\lg n - \lg 2) + 1 \tag{3}\\
&= c(\lg n - 1) + 1 \tag{4}\\
&= c \lg n - c + 1 \tag{5}\\
&\le c \lg n - c + c \tag{6}
\end{align}
$$

Correct Solution:

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

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

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

$$\begin{align} T(n) &\leq c \lg(\lceil n/2 \rceil) + 1 \tag{1}\\ &\leq c \lg(n/2) + 1 \tag{2}\\ &= c(\lg n - \lg 2) + 1 \tag{3}\\ &= c(\lg n - 1) + 1 \tag{4}\\ &= c \lg n - c + 1 \tag{5}\\ &\leq c \lg n - c + c \tag{6}\end{align}$$

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 $T(n) \leq c \cdot n \lg n$.

$$\begin{align} T(n) &= 2T(\lfloor n/2 \rfloor) + n \tag{1}\\ &= 2c\lfloor n/2 \rfloor \lg\lfloor n/2 \rfloor + n \tag{2}\\ &\leq 2c \cdot (n/2) \lg(n/2) + n \tag{3}\\ &= c \cdot n(\lg n - 1) + n \tag{4}\\ &= cn \lg n - cn + n \tag{5}\\ &\leq cn \lg n - cn + cn \tag{6}\\ &\leq cn \lg n \tag{7}\end{align}$$

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

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

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

$$
\begin{aligned}
T(n) &= 2T(\lfloor (n-2)/2 \rfloor) + n && (8)\\
&\ge 2d\lfloor (n-2)/2 \rfloor \lg \lfloor (n-2)/2 \rfloor + n && (9)\\
&\ge 2d \cdot ((n-2)/2)\lg((n-2)/2) + n && (10)\\
&= d \cdot (n-2)(\lg(n-2) - 1) + n && (11)\\
&= d \cdot (n-2)\lg(n-2) - d \cdot (n-2) + n && (12)\\
&\ge d \cdot (n-2)\lg(n-2) - d \cdot (n-2) + (n-2) && (13)\\
&\ge d \cdot (n-2)\lg(n-2) - d \cdot (n-2) + d \cdot (n-2) && (14)\\
&= d \cdot (n-2)\lg(n-2) && (15)
\end{aligned}
$$

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

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 $\boxed{T(n) \le cn \lg n,}$ upper bound

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

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

lower bound

Next, we guess $\boxed{T(n) \ge c(n+2)\lg(n+2),}$

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

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

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

$$\begin{aligned}
T(n) &\leq c(\lceil n/2 \rceil - 2)\lg(\lceil n/2 \rceil - 2) + c(\lfloor n/2 \rfloor - 2)\lg(\lfloor n/2 \rfloor - 2) + dn && (1)\\
&= c(n/2 + 1 - 2)\lg(n/2 + 1 - 2) + c(n/2 + 1 - 2)\lg(n/2 + 1 - 2) + dn && (2)\\
&= c((n-2)/2)\lg((n-2)/2) + c((n-2)/2)\lg((n-2)/2) + dn && (3)\\
&= c(n-2)\lg((n-2)/2) + dn && (4)\\
&= c(n-2)\lg(n-2) - c(n-2) + dn && (5)\\
&= c(n-2)\lg(n-2) - (d-c)n + 2c && (6)\\
&= c(n-2)\lg(n-2) && (7)
\end{aligned}$$

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

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

$$\begin{aligned}
T(n) &\leq c(\lceil n/2 \rceil + 1)\lg(\lceil n/2 \rceil + 1) + c(\lfloor n/2 \rfloor + 1)\lg(\lfloor n/2 \rfloor + 1) + dn && (8)\\
&\leq c(n/2 - 1 + 1)\lg(n/2 - 1 + 1) + c(n/2 - 1 + 1)\lg(n/2 - 1 + 1) + dn && (9)\\
&= c(n/2)\lg(n/2) + c(n/2)\lg(n/2) + dn && (10)\\
&= cn\lg(n/2) + dn && (11)\\
&= cn\lg(n) - cn + dn && (12)\\
&= cn\lg(n) + (d-c)n && (13)\\
&\leq c(n-1)\lg(n-1) && (14)
\end{aligned}$$

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): $cn \lg n$

$$
\begin{aligned}
T(n) &\le 2c(\lfloor n/2 \rfloor + 17)\lg(\lfloor n/2 \rfloor + 17) + n && (15)\\
&\le 2c((n/2) + 17)\lg((n/2) + 17) + n && (16)\\
&= 2c(n/2)\lg(n/2) + n && (17)\\
&= cn(\lg(n) - 1) + n && (18)\\
&= cn\lg(n) - cn + n && (19)\\
&\le cn\lg(n) - cn + cn && (20)\\
&= cn\lg(n) && (21)
\end{aligned}
$$

6.

$$
\begin{aligned}
T(n) &= 4T(n/3) + n && (1)\\
&\le 4c(n/3)^{\log_3 4} + n && (2)\\
&\le 4c(1/3)^{\log_3 4} n^{\log_3 4} + n && (3)\\
&\le (4/4)cn^{\log_3 4} + n && (4)\\
&\le cn^{\log_3 4} + n && (5)
\end{aligned}
$$

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

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

$$
\begin{aligned}
T(n) &= 4T(n/3) + n && (6)\\
&\le 4c(n/3)^{\log_3 4} - d(n/3) + n && (7)\\
&= 4c(n/3)^{\log_3 4} - d(n/3) + n && (8)\\
&= (4/3^{\log_3 4})cn^{\log_3 4} - d(n/3) + n && (9)\\
&= (4/4)cn^{\log_3 4} - d(n/3) + n && (10)\\
&= cn^{\log_3 4} - d(n/3) + n && (11)\\
&\le cn^{\log_3 4} - d(n/3) + n && (12)\\
&\le cn^{\log_3 4} && (13)
\end{aligned}
$$

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

Correct Solution:

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

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

$$
\begin{aligned}
T(n) &\le 2c(\lfloor n/2 \rfloor + 17)\lg(\lfloor n/2 \rfloor + 17) + n && (14)\\
&\le 2c((n/2) + 17)\lg((n/2) + 17) + n && (15)\\
&= 2c(n/2)\lg(n/2) + n && (16)\\
&= cn(\lg(n) - 1) + n && (17)\\
&= cn\lg(n) - cn + n && (18)\\
&\le cn\lg(n) - cn + cn && (19)\\
&= cn\lg(n) && (20)
\end{aligned}
$$

$$
\begin{aligned}
T(n) &= 4T(n/3) + n && (1)\\
&\le 4c(n/3)^{\log_3 4} + n && (2)\\
&\le 4c(1/3)^{\log_3 4} n^{\log_3 4} + n && (3)\\
&\le (4/4)cn^{\log_3 4} + n && (4)\\
&\le cn^{\log_3 4} + n && (5)
\end{aligned}
$$

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

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

$$
\begin{aligned}
T(n) &= 4T(n/3) + n && (6)\\
&\le 4c(n/3)^{\log_3 4} - 4d(n/3)4d(n/3) + n && (7)\\
&= 4d(n/3) = 4c(n/3)^{\log_3 4} - 4d(n/3) + n && (8)\\
&= 4d(n/3) = (4/3^{\log_3 4})cn^{\log_3 4} - 4d(n/3) + n && (9)\\
&= (4/4)cn^{\log_3 4} - 4d(n/3) + n && (10)\\
&= cn^{\log_3 4} - 4d(n/3) + n && (11)\\
&\le cn^{\log_3 4} - 4d(n/3) + n && (12)\\
&\le cn^{\log_3 4} - 4d(n/2) + n && (13)\\
&\le cn^{\log_3 4} - 2dn + n && (14)\\
&\le cn^{\log_3 4} - 2dn + dn && (15)\\
&\le cn^{\log_3 4} - dn && (16)
\end{aligned}
$$

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

$$
\begin{aligned}
T(n) &= 4T(n/2) + n && (17)\\
&\le 4c(n/2)^2 + n && (18)\\
&= (4/4)cn^2 + n && (19)\\
&= cn^2 + n && (20)
\end{aligned}
$$

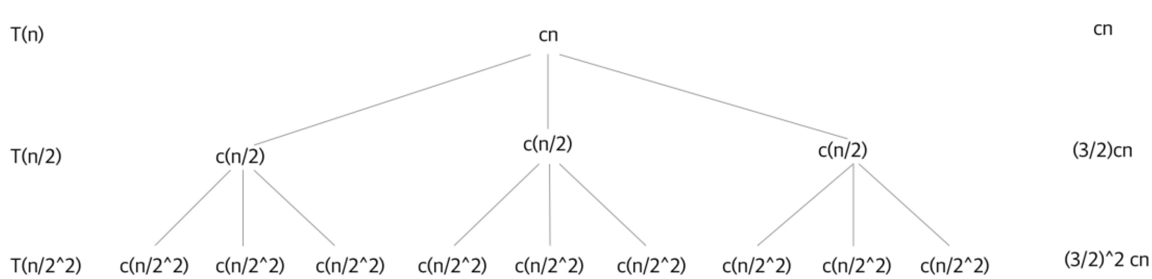
We cannot advance further since n in $cn^2 + n$ cannot be eliminated.

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

$$
\begin{aligned}
T(n) &= 4T(n/2) + n && (21)\\
&\le 4c(n/2)^2 - 4d(n/2) + n && (22)\\
&= (4/4)cn^2 - 2dn + n && (23)\\
&\le cn^2 - 2dn + dn && (24)\\
&= cn^2 - dn && (25)
\end{aligned}
$$

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

8. **Solution:**



1. Finding number of levels in recursion tree

$$
\begin{aligned}
1 &= n/2^i && (1)\\
2^i &= n && (2)\\
i &= \log_2 n && (3)
\end{aligned}
$$

2. Finding the total cost of recursion tree

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

$$
\begin{align}
T(n) &= n \cdot \sum_{i=0}^{\lg_2(n)-1} (3/2)^i + \Theta(n^{\lg 3}) \tag{4}\\
&= n \cdot \Big(\frac{(3/2)^{\lg_2(n)} - 1}{(3/2) - 1}\Big) + \Theta(n^{\lg 3}) \tag{5}\\
&= 2n \cdot \Big((3/2)^{\lg_2(n)} - 1\Big) + \Theta(n^{\lg 3}) \tag{6}\\
&= 2n \cdot \Big(n^{\lg(3/2)} - 1\Big) + \Theta(n^{\lg 3}) \tag{7}\\
&= 2n \cdot \Big(n^{\lg(3/2)} - 1\Big) + \Theta(n^{\lg 3}) \tag{8}\\
&= 2 \cdot \Big(n^{\lg 3 - 1 + 1} - n\Big) + \Theta(n^{\lg 3}) \tag{9}\\
&= 2 \cdot \Big(n^{\lg 3} - n\Big) + \Theta(n^{\lg 3}) \tag{10}\\
&= 2 \cdot \Big(n^{\lg 3} - n\Big) + \Theta(n^{\lg 3}) \tag{11}
\end{align}
$$

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

3. Verifying the correct guess using the subtitution method

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

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

   Indeed we have

$$
\begin{align}
T(n) &= 3T(\lfloor n/2 \rfloor) + n \tag{12}\\
&\leq 3(c\lfloor n/2 \rfloor^{\lg 3}) - d(\lfloor n/2 \rfloor) + n \tag{13}\\
&= 3\Big(\frac{cn^{\lg 3}}{3} - d(\frac{n}{2} + 1)\Big) + n \tag{14}\\
&= 3\Big(\frac{cn^{\lg 3}}{3} - \frac{3dn}{2}\Big) + n \tag{15}\\
&\leq 3\Big(\frac{cn^{\lg 3}}{3} - \frac{3dn}{3}\Big) + n \tag{16}\\
&= cn^{\lg 3} - 3dn + n \tag{17}\\
&\leq cn^{\lg 3} - 3dn + 2dn \tag{18}\\
&= cn^{\lg 3} - dn \tag{19}
\end{align}
$$

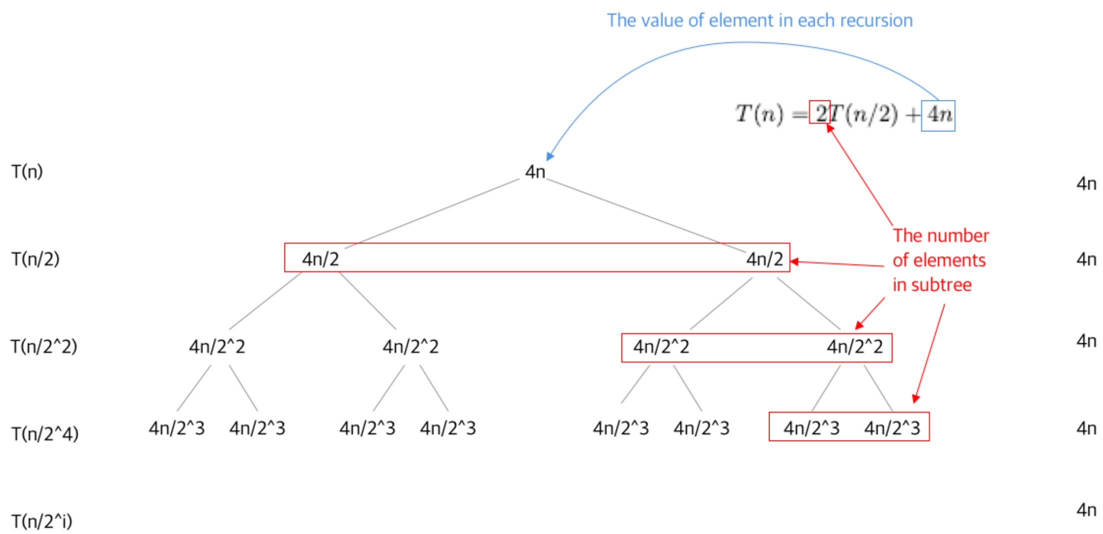
And the boundary holds as long as $c \geq 0$ and $d \geq 1$.

Notes:

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

Example:

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



1. Finding number of levels in recursion tree

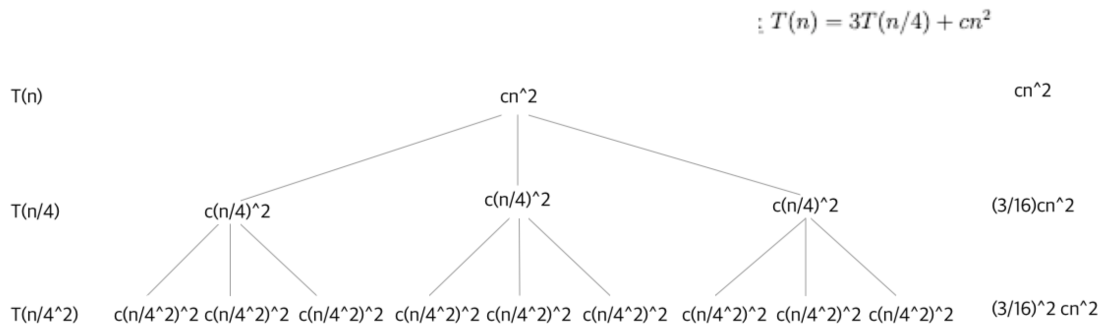
$$1 = n/2^i \tag{20}$$
$$2^i = n \tag{21}$$
$$i = \log_2 n \tag{22}$$

2. Finding the value of guess

$$\sum_{i=0}^{\log_2 n} 4n = 4n \cdot \sum_{i=0}^{\log_2 n} 1 \tag{23}$$
$$= 4n(\log_2 n + 1) \tag{24}$$

Example 2:

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



Steps:

1. Finding number of levels in recursion tree

$$1 = n/4^i \tag{25}$$
$$4^i = n \tag{26}$$
$$i = \log_4 n \tag{27}$$

2. Finding the cost of entire tree

$$T(n) = \sum_{i=0}^{\log_4 n-1} c(3/16)^i n^2 + \Theta(n^{\log_4 3}) \tag{28}$$
$$= cn^2 \cdot \sum_{i=0}^{\log_4 n-1} (3/16)^i + \Theta(n^{\log_4 3}) \tag{29}$$
$$< cn^2 \cdot \sum_{i=0}^{\infty} (3/16)^i + \Theta(n^{\log_4 3}) \qquad \text{[since } n \text{ is asympt. large]} \tag{30}$$
$$= cn^2\left(\frac{1}{1-(3/16)}\right) + \Theta(n^{\log_4 3}) \qquad \text{[Since } \sum_{i=0}^{\infty} ar^i = \frac{a}{1-r}\text{]} \tag{31}$$

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

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

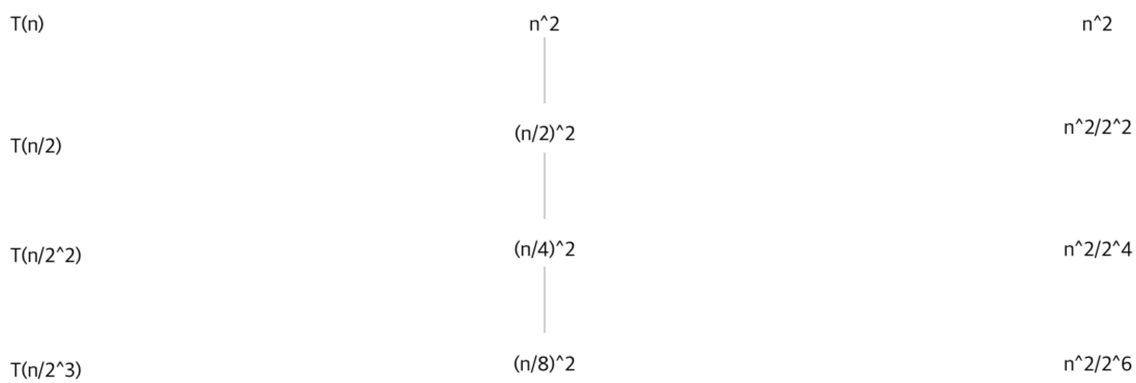
   Since the total cost is $T(n) = cn^2\left(\frac{1}{1-(3/16)}\right) + \Theta(n^{\log_4 3})$, we have $\mathcal{O}(n^2)$

4. Verify the correctness of guess using subtitution method

$$
\begin{aligned}
T(n) &\leq 3T(\lfloor n/4 \rfloor) + cn^2 && (32)\\
&\leq 3d\lfloor n/4 \rfloor^2 + cn^2 && (33)\\
&\leq 3d(n/4)^2 + cn^2 && (34)\\
&= (3/16)dn^2 + cn^2 && (35)\\
&\leq dn^2 && (36)
\end{aligned}
$$

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

9. **Solution:**



1. Finding number of levels in recursion tree

$$
\begin{aligned}
1 &= \frac{n}{2^i} && (1)\\
2^i &= n && (2)\\
i &= \lg n && (3)
\end{aligned}
$$

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

$$
\begin{aligned}
T(n) &= n^2 \cdot \sum_{i=0}^{\lg n - 1} \frac{1}{2^{2i}} + \Theta(1) && (4)\\
&= n^2 \cdot \sum_{i=0}^{\infty} \frac{1}{2^{2i}} + \Theta(1) && \text{[since } n \text{ is asympt. large]} \quad (5)\\
&= n^2 \cdot \left(\frac{1}{1-\frac{1}{4}}\right) + \Theta(1) && (6)\\
&= \frac{4n^2}{3} + \Theta(1) && (7)
\end{aligned}
$$

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

3. Verify the correctness of guess using subtitution method

   <u>Guess:</u> $T(n) \leq cn^2$

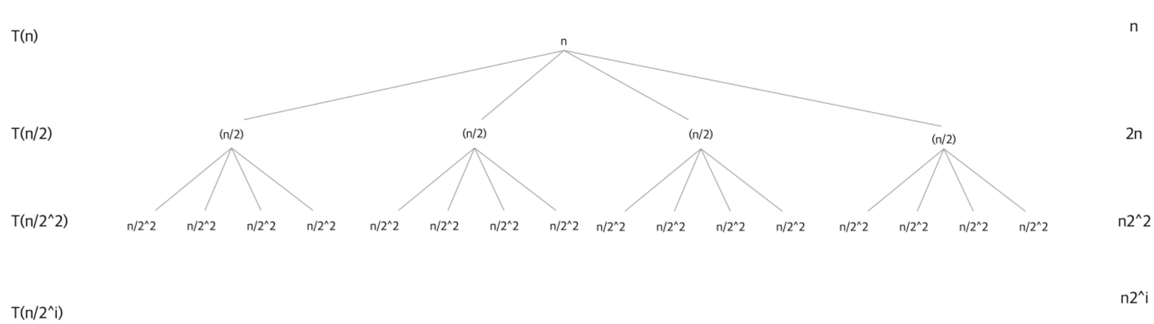
   I need to show the guess holds for the recurrence $T(n) = T(\frac{n}{2}) + n$.

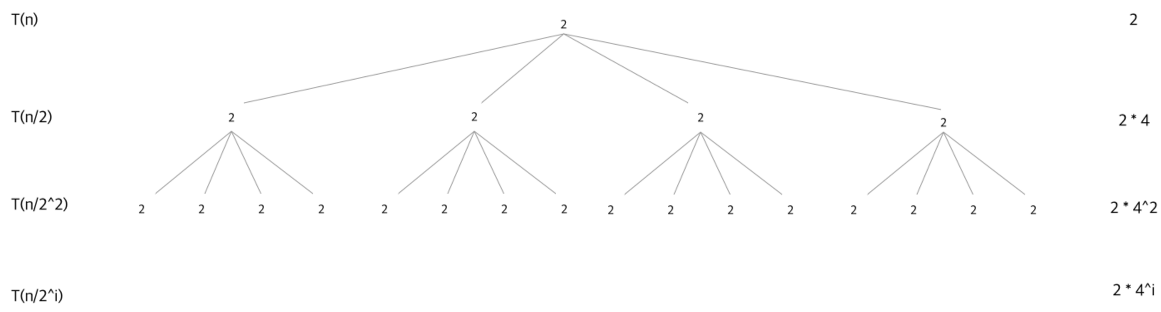
   And, indeed we have

$$
\begin{aligned}
T(n) &= T(\frac{n}{2}) + n^2 && (8)\\
&\leq \frac{cn^2}{4} + n^2 && (9)\\
&= (\frac{c}{4} + 1) \cdot n^2 && (10)\\
&\leq cn^2 && (11)
\end{aligned}
$$

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

10. **<u>Solution:</u>**





- Find the total cost of the recursion tree

$$1 = \frac{n}{2^i} \tag{1}$$
$$2^i = n \tag{2}$$
$$i = \lg n \tag{3}$$

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

$$\begin{aligned}
T(n) &= \sum_{i=0}^{\lg n - 1} (2 \cdot 4^i + n2^i) + \Theta(n^2) && (4)\\
&= \sum_{i=0}^{\lg n - 1} 2 \cdot 4^i + \sum_{i=0}^{\lg n - 1} n2^i + \Theta(n^2) && (5)\\
&= 2 \cdot \sum_{i=0}^{\lg n - 1} 4^i + n \cdot \sum_{i=0}^{\lg n - 1} 2^i + \Theta(n^2) && (6)\\
&= 2 \cdot \left(\frac{4^{\lg n} - 1}{4 - 1}\right) + n \cdot (n-1) + \Theta(n^2) \quad \left[\text{Since } \sum_{i=0}^{n-1} ar^i = a \cdot \frac{r^n - 1}{r - 1}, \text{ where } r \neq 1\right] && (7)\\
&= \frac{2}{3} \cdot (n^2 - 1) + n \cdot (n-1) + \Theta(n^2) && (8)\\
&= \mathcal{O}(n^2) + \Theta(n^2) && (9)\\
& && (10)
\end{aligned}$$

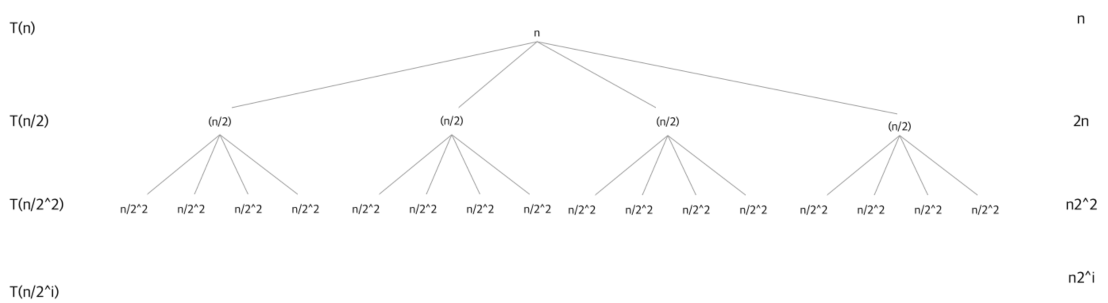
- Verify the correctness of guess using subtitution method

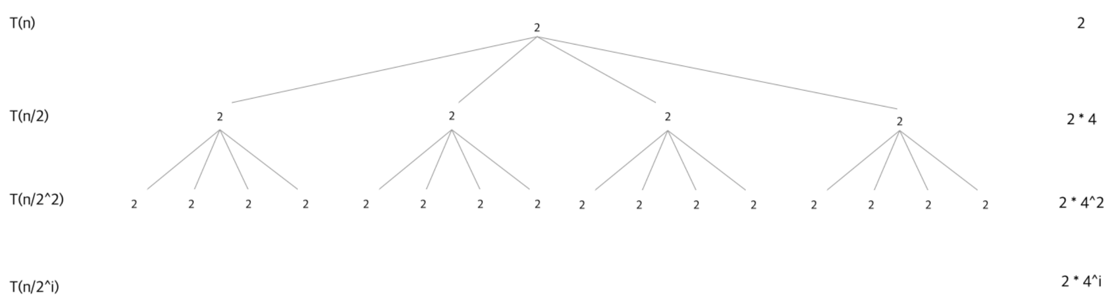
  Guess: $T(n) \leq cn^2 - dn$.

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

$$
\begin{aligned}
T(n) &= 4T(\frac{n}{2}+2)+n && (11)\\
&\le 4c(\frac{n}{2}+2)^2 - 4dn + n && (12)\\
&= 4c(\frac{n^2}{4}+2n+4) - 4dn + n && (13)\\
&\le cn^2 - 4dn + n \qquad \text{[Since } n^2 \text{ dominates } n \text{ asymptotically]} && (14)\\
&\le cn^2 - 4dn + 3dn && (15)\\
&= cn^2 - dn && (16)
\end{aligned}
$$

Correct Solution:





- Find the total cost of the recursion tree

$$
\begin{aligned}
1 &= \frac{n}{2^i} && (17)\\
2^i &= n && (18)\\
i &= \lg n && (19)
\end{aligned}
$$

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

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\lg n-1} (2 \cdot 4^i + n2^i) + \Theta(n^2) && (20)\\
&= \sum_{i=0}^{\lg n-1} 2 \cdot 4^i + \sum_{i=0}^{\lg n-1} n2^i + \Theta(n^2) && (21)\\
&= 2 \cdot \sum_{i=0}^{\lg n-1} 4^i + n \cdot \sum_{i=0}^{\lg n-1} 2^i + \Theta(n^2) && (22)\\
&= 2 \cdot \left(\frac{4^{\lg n} - 1}{4-1}\right) + n \cdot (n-1) + \Theta(n^2) \quad \left[\text{Since } \sum_{i=0}^{n-1} ar^i = a \cdot \frac{r^n - 1}{r-1}, \text{ where } r \neq 1\right] && (23)\\
&= \frac{2}{3} \cdot (n^2 - 1) + n \cdot (n-1) + \Theta(n^2) && (24)\\
&= \Theta(n^2) && (25)
\end{aligned}
$$

- Verify the correctness of guess using subtitution method

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

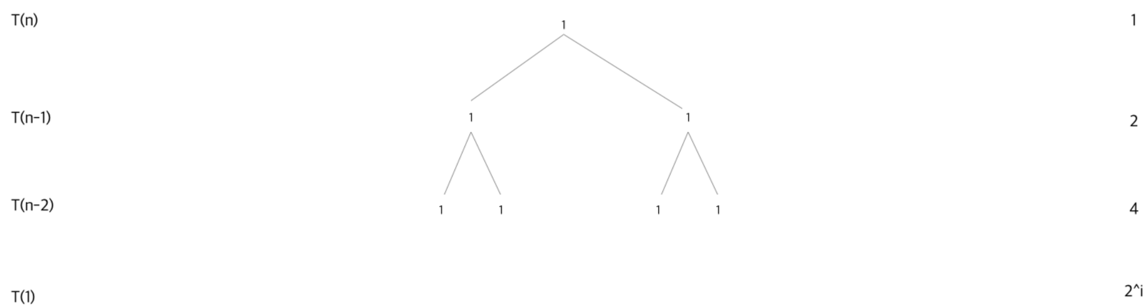
  I need to show the guess holds for the recurrence $T(n) = 4T(\frac{n}{2} + 2) + n$

$$
\begin{aligned}
T(n) &= 4T(\frac{n}{2} + 2) + n && (26)\\
&\leq 4c(\frac{n}{2} + 2)^2 - 4dn + n && (27)\\
&= 4c(\frac{n^2}{4} + 2n + 4) - 4dn + n && (28)\\
&\leq cn^2 - 4dn + n \quad [\text{Since } n^2 \text{ dominates } n \text{ asymptotically}] && (29)\\
&\leq cn^2 - 4dn + 3dn && (30)\\
&= cn^2 - dn && (31)
\end{aligned}
$$

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? I 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

$$n - 1 \tag{1}$$

- Finding the number of leaves in the tree

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

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

$$\begin{aligned} T(n) &\leq \sum_{i=0}^{n-1} 2^i + \Theta(2^n) && (3)\\ &= \left(\frac{2^n - 1}{2 - 1}\right) + \Theta(2^n) && (4)\\ &= (2^n - 1) + \Theta(2^n) && (5)\\ &= \Theta(2^n) && (6) \end{aligned}$$

- Verify the correctness of guess using subtitution method

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

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

  Indeed we have

$$
\begin{aligned}
T(n) &= 2T(n-1) + 1 && (7)\\
&< 2c2^{n-1} + 1 && (8)\\
&= c2^{n} + 1 && (9)\\
&= c2^{n} && \text{[Since } n \text{ is asympt. large]} \quad (10)
\end{aligned}
$$

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

$$\text{number of branchings}^{\text{depth of tree}} \tag{11}$$

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

12. **Solution:**

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:

$$\text{number of branchings}^{\text{depth of tree}} = 2^{2^{n-1}} \tag{12}$$

3. Find the upper bound of $T(...)$ that produces most depth
4. Valiate the upper bound using the subtitution method

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 $T(...)$ that produces most depth

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

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

4. Valiate the upper bound using the subtitution method

$$\begin{aligned}
T(n) &\leq T(\frac{n}{3}) + T(\frac{2n}{3}) + cn && (13)\\
&\leq d(\frac{n}{3}) \cdot \lg(\frac{n}{3}) + d(\frac{2n}{3}) \lg(\frac{2n}{3}) + cn && (14)\\
&= (d(\frac{n}{3}) \lg n - d(\frac{n}{3}) \cdot \lg 3) + (d(\frac{2n}{3}) \lg n - d(\frac{2n}{3}) \lg(\frac{3}{2}) + cn) && (15)\\
&= dn \lg n - d((\frac{n}{3} \lg 3) + (\frac{2n}{3}) \lg(3/2)) + cn && (16)\\
&= dn \lg n - d((\frac{n}{3}) \lg 3 + (\frac{2n}{3}) \lg(3) - (\frac{2n}{3}) \lg(2)) + cn && (17)\\
&= dn \lg n - dn(\lg 3 - \frac{2}{3}) + cn && (18)\\
&\leq dn \lg n && (19)
\end{aligned}$$

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