

Assignment 2

Solving Recurrences

Name: Dipam Sen
Roll No: 24CS10059

2.4.1 Using the substitution method, show that the solution of the recurrence $T(n) = T(n-1) + n$ is $O(n^2)$. Establish it also using recursion tree method.

Sol. Substitution Method

Assume **inductive hypothesis**: $T(k) \leq ck^2$ for some $c > 0$ for all $k < n$.

Then,

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

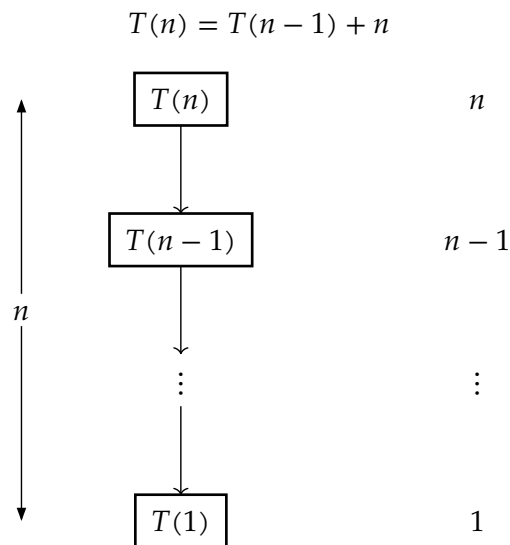
Step:

$$\begin{aligned} T(n) &\leq c(n-1)^2 + n \\ &= cn^2 - 2cn + c + n \\ &\leq cn^2 \quad \text{for } c \geq 1, n > 1 \end{aligned}$$

Suitable basis condition is $c = 1, n_0 = 2$

This proves that $T(n) = O(n^2)$.

Recursion Tree Method



At each level of the recursion tree, input size decreases by 1. Hence the number of levels is n . Each level has exactly one node and work done at k th node is $n - k$, so total time across each level is $\sum_{i=1}^n i = \frac{n(n+1)}{2} \Rightarrow T(n) = O(n^2)$.

2.4.2 Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$.

Sol.

Claim:

$$n_k = \underbrace{\lceil \dots \lceil n/2 \rceil \dots /2 \rceil}_{k \text{ ceilings}} = \lceil n/2^k \rceil$$

Proof of claim: (Induction over k)

Basis: For $k = 1$, statement is trivially true. $\lceil n/2 \rceil = \lceil n/2^1 \rceil$.

Step: Assume that the statement is true for $k = m - 1$. Then $n_{m-1} = \underbrace{\lceil \dots \lceil n/2 \rceil \dots /2 \rceil}_{m-1 \text{ ceilings}} = \lceil n/2^{m-1} \rceil$

For $k = m$,

$$n_m = \underbrace{\lceil \dots \lceil n/2 \rceil \dots /2 \rceil}_{m \text{ ceilings}} = \lceil n_{m-1}/2 \rceil = \lceil \lceil n/2^{m-1} \rceil /2 \rceil$$

Let us take $2^{m-1} = p$. We need to simplify $\lceil \lceil n/p \rceil /2 \rceil$.

Let us write n in terms of $2p$. We can write $n = 2pq + r$, for $0 \leq r < 2p$, with unique $q, r \in \mathbb{Z}$.

Case 1: $r = 0$

$$\lceil \lceil n/p \rceil /2 \rceil = \lceil \lceil (2pq)/p \rceil /2 \rceil = \lceil \lceil 2q \rceil /2 \rceil = q$$

Case 2: $0 < r \leq p$

$$\lceil \lceil n/p \rceil /2 \rceil = \lceil \lceil (2pq + r)/p \rceil /2 \rceil = \lceil \lceil 2q + r/p \rceil /2 \rceil$$

Here $0 < \frac{r}{p} \leq 1$, so we get $\lceil (2q + 1)/2 \rceil = \lceil q + 1/2 \rceil = q + 1$.

Case 3: $p < r < 2p$

$$\lceil \lceil n/p \rceil /2 \rceil = \lceil \lceil (2pq + r)/p \rceil /2 \rceil = \lceil \lceil 2q + r/p \rceil /2 \rceil$$

Here $1 < \frac{r}{p} < 2$, so we get $\lceil (2q + 2)/2 \rceil = \lceil q + 1 \rceil = q + 1$.

Thus,

$$\lceil \lceil n/p \rceil /2 \rceil = \begin{cases} q & r = 0 \\ q + 1 & r \neq 0 \end{cases} \quad (1.1)$$

Notice that this is equivalent to the expression $\lceil n/2p \rceil$:

Case 1: $r = 0$

$$\lceil n/2p \rceil = \lceil 2pq/2p \rceil = q$$

Case 2: $0 < r \leq 2p$

$$\lceil n/2p \rceil = \lceil (2pq + r)/2p \rceil = \lceil q + r/2p \rceil$$

Here $0 < \frac{r}{2p} \leq 1$, so we get $q + 1$.

Thus we have

$$\lceil n/2p \rceil = \begin{cases} q & r = 0 \\ q + 1 & r \neq 0 \end{cases} \quad (1.2)$$

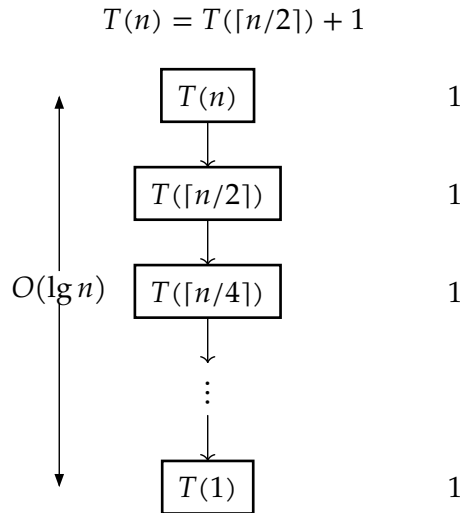
From (1.1) and (1.2) we can say

$$\lceil \lceil n/p \rceil / 2 \rceil = \lceil n/2p \rceil$$

For $p = 2^{m-1}$, we have $\lceil \lceil n/2^{m-1} \rceil / 2 \rceil = \lceil n/2^m \rceil$. This proves our inductive step and thus completes the induction. Thus our claim is verified.

Let's solve the given recurrence using **Recursion Tree Method**.

By using the **claim**, we can simplify the recurrence nodes. At k th level, the recursive step will be $T(\underbrace{\lceil \dots \lceil n/2 \rceil \dots \rceil}_{k \text{ ceilings}}) = T(n_k) = T(\lceil n/2^k \rceil)$.



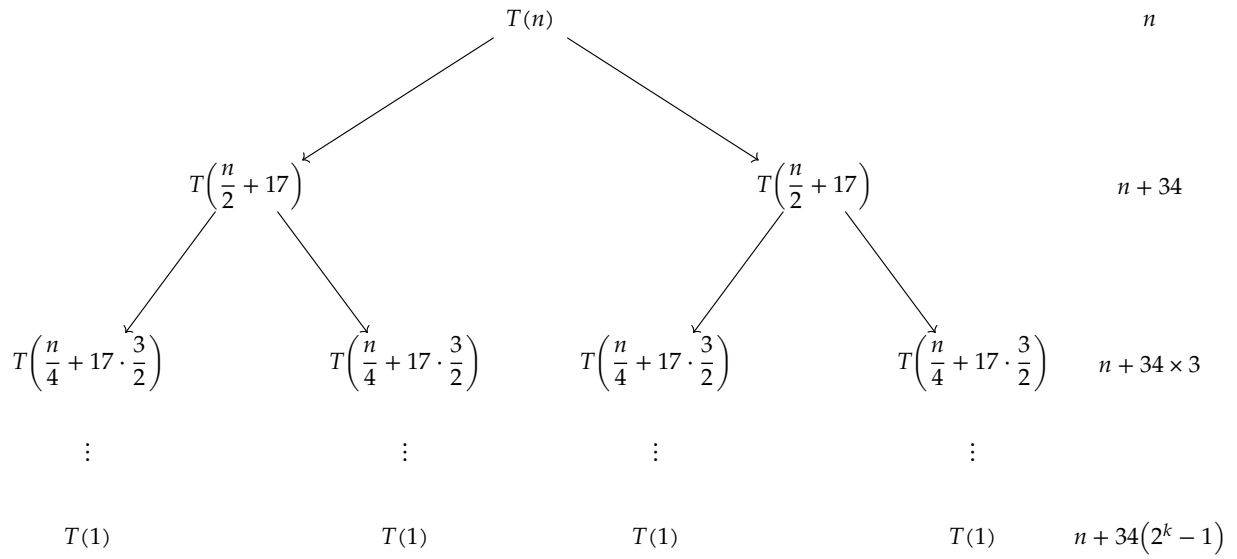
Each level has one node, and extra work at each level is constant. Thus, the height of the tree itself gives the total work, which is $O(\lg n)$.

$$T(n) = O(\lg n)$$

2.4.3 Using Recursion Tree Method, show that $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ is $O(n \lg n)$.

Sol. Since we need to find the upper bound, we can approximate $\lfloor n/2 \rfloor$ as $\frac{n}{2}$, and this won't change the asymptotic behaviour of the recurrence.

$$T(n) = 2T\left(\frac{n}{2} + 17\right) + n$$



Work at k th level:

$$\begin{aligned}
 &= 2^k \times \left(\frac{n}{2^k} + 17 \times \left(1 + \frac{1}{2} + \dots + \frac{1}{2^{k-1}} \right) \right) \\
 &= n + 17 \times (2 + 4 + \dots + 2^k) \\
 &= n + 34 \times (1 + 2 + 4 + \dots + 2^{k-1}) = n + 34 \times (2^k - 1)
 \end{aligned}$$

Total levels = $\Theta(\lg n)$.

Summing up all the works for each level, we have

$$\begin{aligned}
 T(n) &= \sum_{k=1}^{\lg n} n + \sum_{k=1}^{\lg n} 34(2^k - 1) \\
 &= n \lg n + O(2^{\lg(n)}) \\
 &= n \lg n + O(n) \\
 \Rightarrow T(n) &= O(n \lg n)
 \end{aligned}$$

2.4.4 Show that a substitution proof for $T(n) = 4T\left(\frac{n}{3}\right) + n$ fails with $T(n) \leq cn^{\log_3 4}$, and how to fix it by subtracting a lower order term.

Sol.

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

Hypothesis $T(m) \leq cm^{\log_3 4}$ for all $m < n$.

Step

$$\begin{aligned}
T(n) &= 4T\left(\frac{n}{3}\right) + n \\
&\leq 4c\left(\frac{n}{3}\right)^{\log_3 4} + n \\
&= cn^{\log_3 4} + n
\end{aligned}$$

This way, we cannot show $T(n) \leq cn^{\log_3 4}$.

We can fix it by making a stronger inductive hypothesis, by subtracting a lower order term.

Hypothesis $T(m) \leq cm^{\log_3 4} - dm$ for all $m < n$.

Step

$$\begin{aligned}
T(n) &= 4T\left(\frac{n}{3}\right) + n \\
&\leq 4c\left(\frac{n}{3}\right)^{\log_3 4} - 4d\left(\frac{n}{3}\right) + n \\
&= cn^{\log_3 4} - \frac{4}{3}dn + n \\
&= cn^{\log_3 4} - dn - n\left(\frac{1}{3}d - 1\right) \\
&\leq cn^{\log_3 4} - dn \quad \text{for } d > 3
\end{aligned}$$

This completes the proof.

2.4.5 Solve the recurrence $T(n) = T(\sqrt{n}) + \lg n$ by making a change of variables. Give an asymptotically tight bound. Do not worry about whether values are integral.

Sol. Assume $n = 2^k$. Then we have

$$\begin{aligned}
T(2^k) &= T(\sqrt{2^k}) + \lg 2^k \\
&\Rightarrow T(2^k) = T(2^{\frac{k}{2}}) + k
\end{aligned}$$

Define a new function $S(k) = T(2^k)$.

$$S(k) = S\left(\frac{k}{2}\right) + k$$

Clearly, $S(k) \geq k$.

If we use recursion tree method, we find that

$$S(k) = k + \frac{k}{2} + \dots + 1 \leq 2k$$

Together, this implies $S(k) = \Theta(k)$.

But $S(k) = T(2^k) = \Theta(k)$.

Put $k = \lg n$

$$\Rightarrow T(n) = \Theta(\lg n)$$

2.4.5 Solve the recurrence $T(n) = 2T(\sqrt{n}) + \lg n$ by making a change of variables. Give an asymptotically tight bound. Do not worry about whether values are integral.

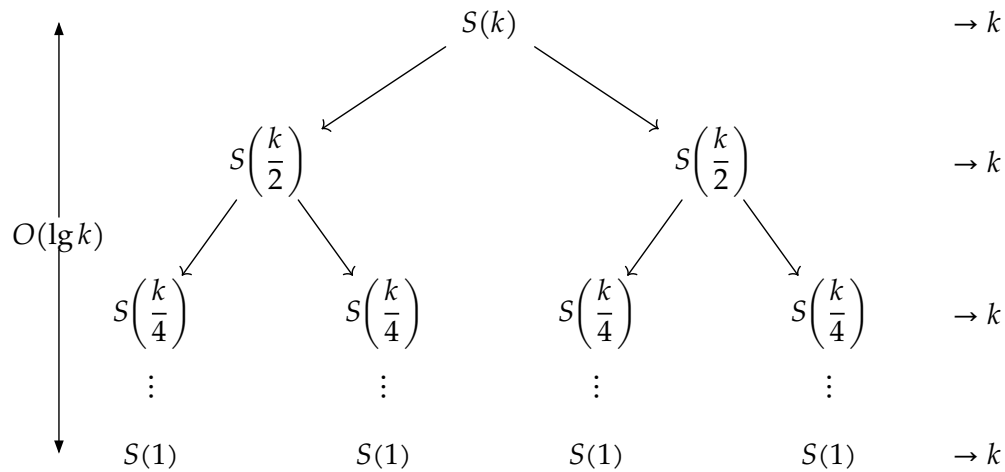
Sol. Assume $n = 2^k$. Then we have

$$T(2^k) = 2T(\sqrt{2^k}) + \lg 2^k$$

$$\Rightarrow T(2^k) = 2T(2^{\frac{k}{2}}) + k$$

Define a new function $S(k) = T(2^k)$.

$$S(k) = 2S\left(\frac{k}{2}\right) + k$$



By recursion tree, we can see that

$$S(k) = \Theta(k \lg k)$$

This is because, at each level total work will be $\Theta(k)$, and there will be $\lg k$ levels.

But $S(k) = T(2^k) = \Theta(k \lg k)$.

Put $k = \lg n$

$$\Rightarrow T(n) = \Theta(\lg n \cdot \lg \lg n)$$