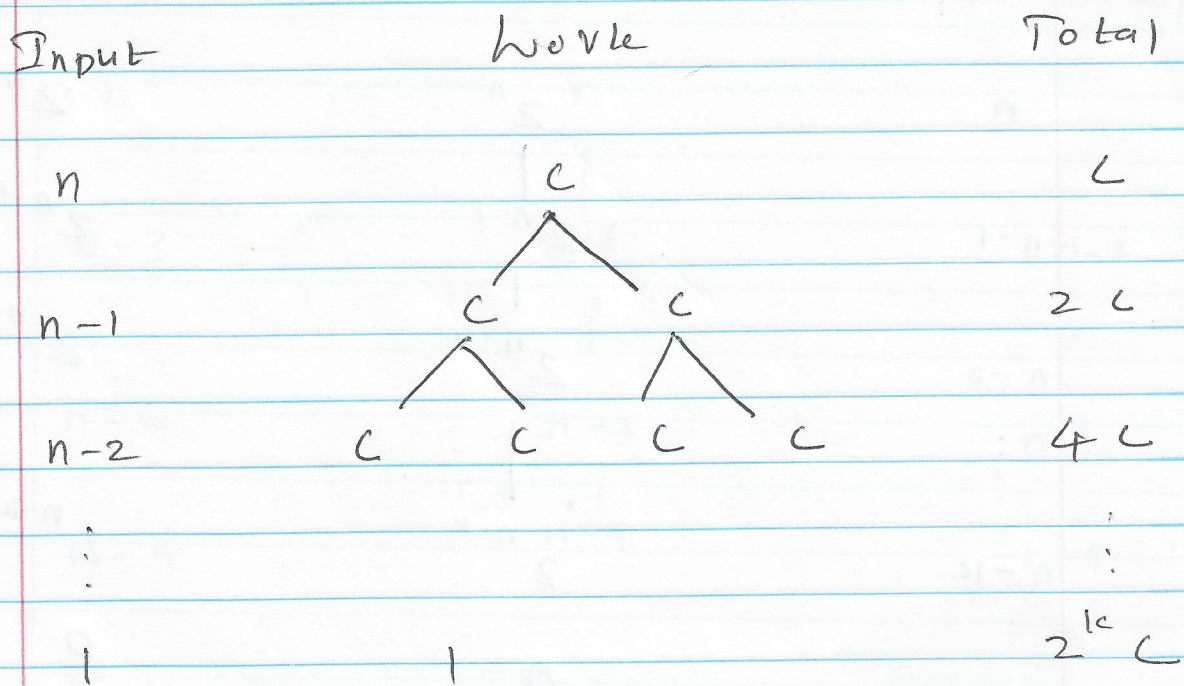


a.) $T(n) = C + 2T(n-1), T(1) = C$

Recursion Tree Method:



Since $T(1) = C$, the recursion ends at $k = n-1$
Hence

$$T(n) = C + 2C + 4C + \dots + C \cdot 2^{(n-1)}$$

$$= C(1 + 2 + 4 + \dots + 2^{(n-1)})$$

$$= C(2^{n-1+1} - 1)$$

$$= C(2^n - 1) \quad \text{Via Summation of Series}$$

$$T(n) = O(2^n)$$

6.) $T(n) = T(n-3) + n, T(0) = 0$ (n multiple of 3)

Recursion tree method

Input	Work	Total
n	n	n
$n-3$	$n-3$	$n-3$
$n-6$	$n-6$	$n-6$
$n-9$	$n-9$	$n-9$
\vdots	\vdots	\vdots
$n-k$	$n-k$	$n-k$
\vdots	\vdots	\vdots
0	0	0

$$T(n) = n + n-3 + n-6 + \dots + n-k \text{ to } 0$$

Since $T(0) = 0$, Consider $n=k$, $k=n$ & divisible of 3

$$T(n) = n + n-3 + n-6 + \dots + n-n+3 + n-n$$

$$= n + n-3 + n-6 + \dots + 3 + 0$$

$$= \frac{n}{2} (0 + (n-1)(3)) \text{ By Summation of A.P}$$

$$= \frac{3}{2} (n^2 - n)$$

$$T(n) = O(n^2)$$

$$(c) \quad T(n) = 2^n + T(n-1), \quad T(1) = 0$$

Recursion Tree method:

Input	Work	Total
n	2^n	2^n
n-1	2^{n-1}	2^{n-1}
n-2	2^{n-2}	2^{n-2}
⋮	⋮	⋮
n-k	2^{n-k}	2^{n-k}
⋮	⋮	⋮
1	0	0

$$T(n) = 2^n + 2^{n-1} + 2^{n-2} + \dots + 2^{n-k} + 0$$

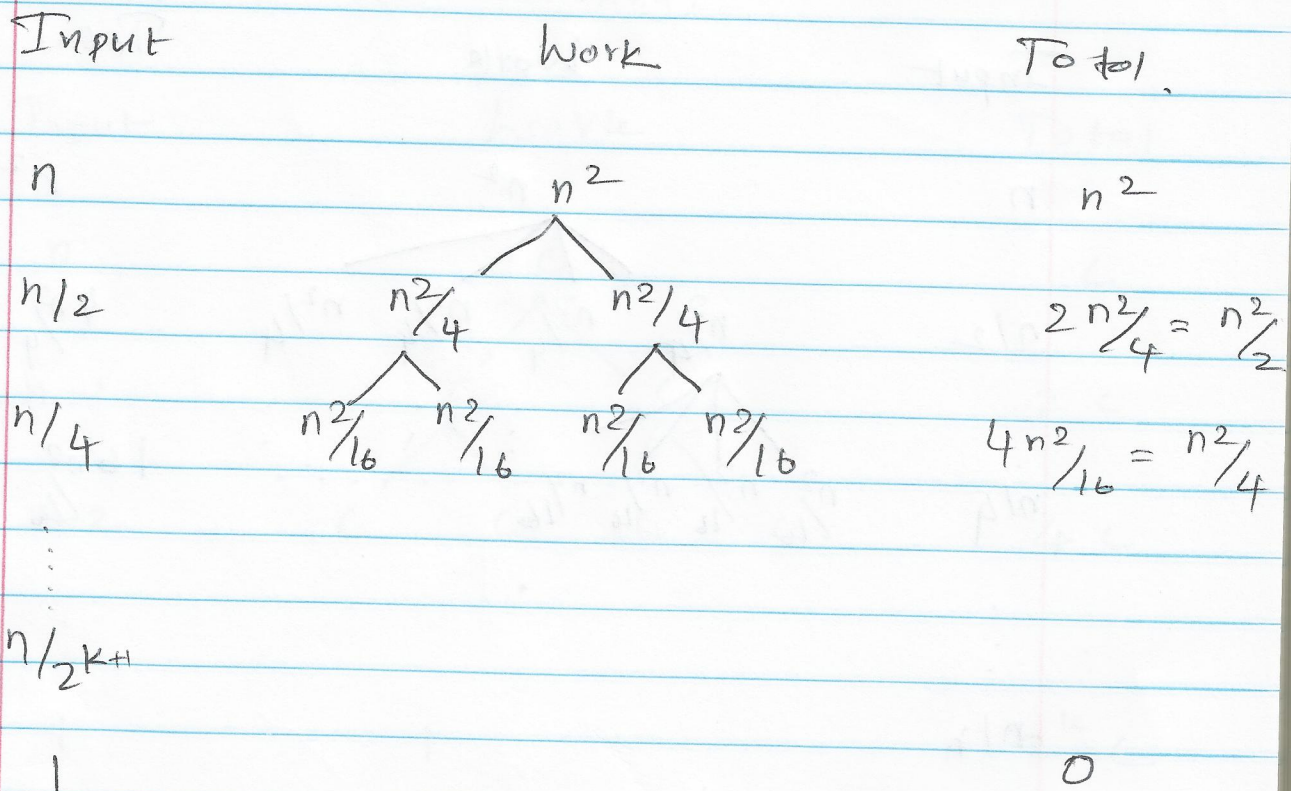
$$= 2^n \left(1 + \frac{1}{2^1} + \frac{1}{2^2} + \dots + \frac{1}{2^k} \right)$$

= Cannot be > 2

$$T(n) = O(2^n)$$

d.) $T(n) = n^2 + 2T(n/2)$, $T(1) = 0$

Recursion Tree Method:



Here $n = 2^{k+1}$ $k = \log n$

$$T(n) = n^2 + n^2/2 + n^2/4 + \dots + n^2/2^k$$

$$= n^2 \left(1 + 1/2 + 1/4 + \dots + 1/2^k \right)$$

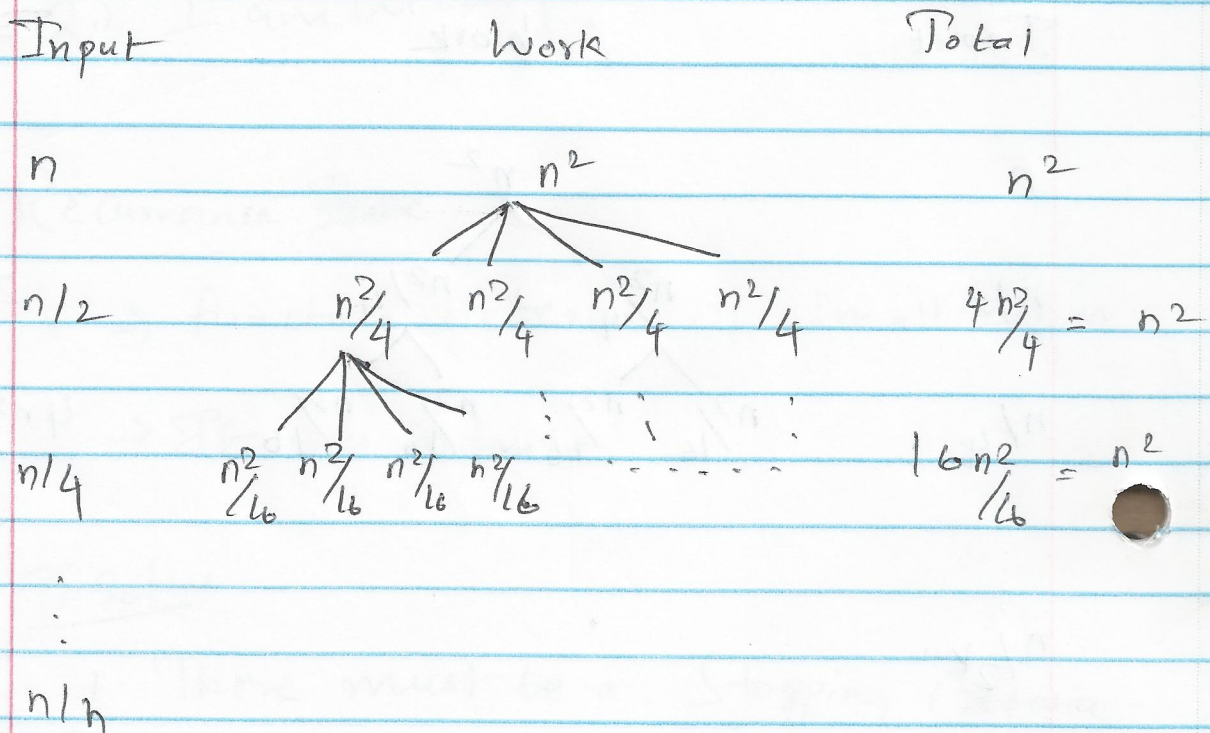
cannot be more than 2

$$T(n) = n^2(2)$$

$$T(n) = O(n^2) \text{ where } C=2$$

e-) $T(n) = n^2 + 4T(n/2)$

Recursion Tree method:



Here $n = 2^{k+1}$ Since ~~n=0~~, When $n=0$, it must be 2^0

$$\begin{aligned}
 T(n) &= n^2 + n^2 + n^2 + \dots + n^2_{2^{k+1}} \\
 &= n^2 (\log n \text{ times})
 \end{aligned}$$

Hence $T(n) = \Theta(n^2 \log n)$