Topics to discuss

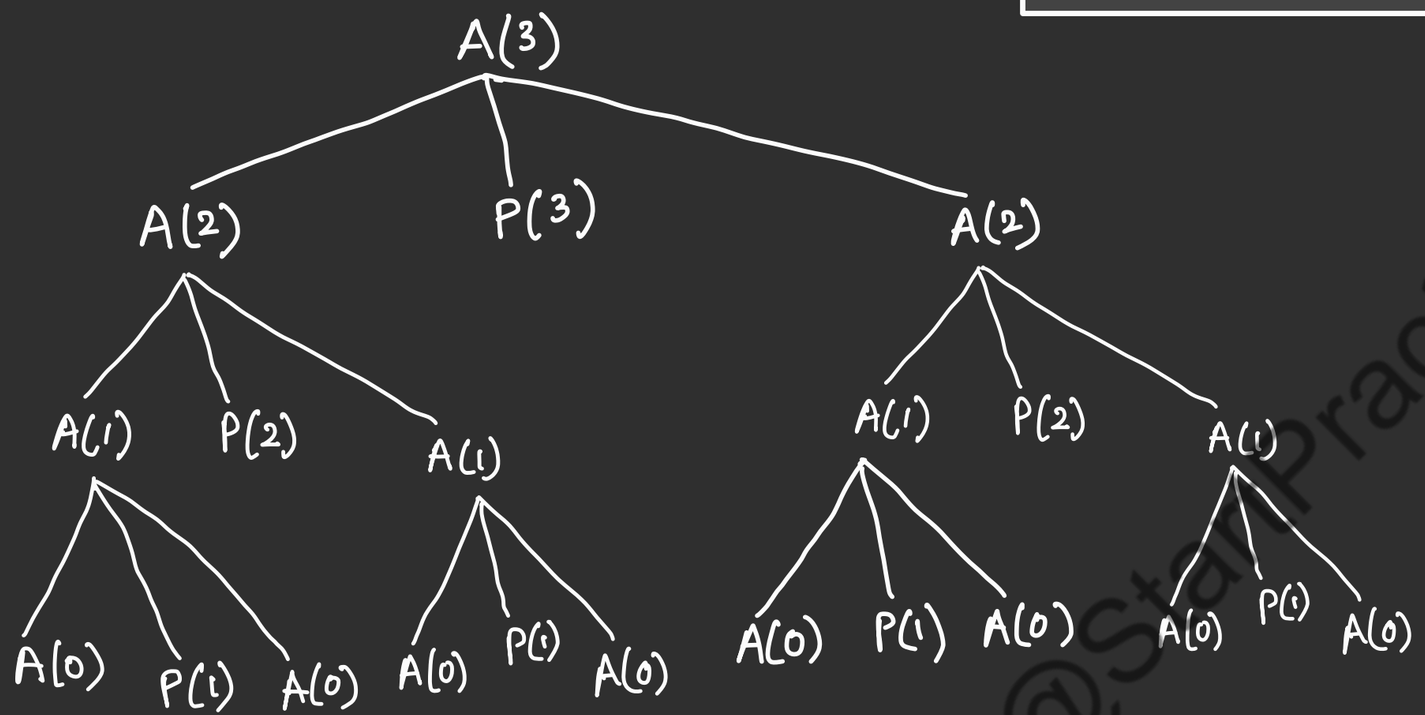
How to find space complexity for recursive algorithm.

Example - 02

Assume, $n = 3$

$$T(n) = \begin{cases} 2T(n-1)+1 & , n \geq 1 \\ 1 & , n = 0 \end{cases}$$

```
A(n)
{
  if(n≥1){
    A(n-1);
    Print (n);
    A(n-1);
    }
}
```

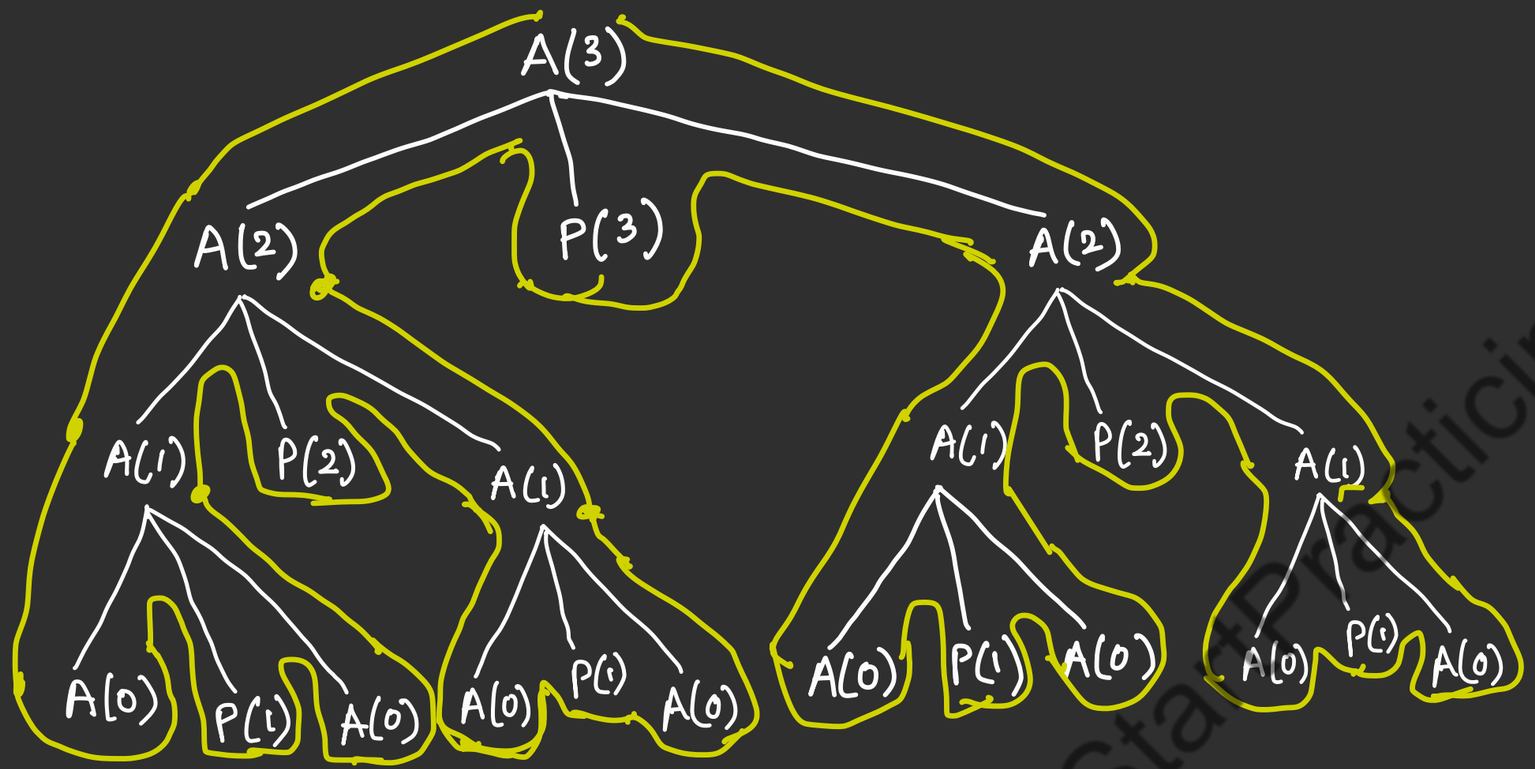


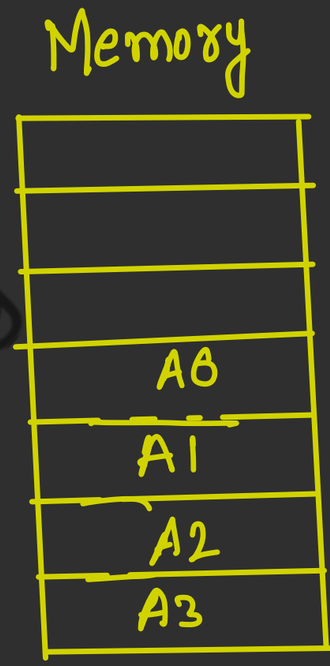
$n = 3 \Rightarrow 15 \Rightarrow 2^{3+1} - 1$

$n = 2 \Rightarrow 7 \Rightarrow 2^{2+1} - 1$

$n = 1 \Rightarrow 3 \Rightarrow 2^{1+1} - 1$

for n, $2^{n+1} - 1$ Calls





1213121

Traverse → Top to down
left to right

$S.C = (n+1)K$

$= nk$

$S.C = O(n)$

Follow Now

Start Practicing

i._am._arfin

Arfin Parween