3 Mar 2020 Date: RF-1

Recursive functions

```
A(n) {
  if ( )
    return (A(n/2) + A(n/2))
}
```

Nothing to count

— Assume time T(n)

— Some const. time

Some const. time

So, $T(n) = c + 2T(n/2)$ — Recursive func.

```
A(n) {
  if (n>1)
    return (A(n-1))
}
```

c or 1

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

Back Substitution — Slow

$T(n) = 1 + T(n-1)$ — ①

$T(n-1) = 1 + T(n-2)$ — ②

$T(n-2) = 1 + T(n-3)$ — ③

Base/Anchor Condition

$T(1)$

$n-k=1$

$k=n-1$

Substitute ② in ①

$T(n) = 1 + 1 + T(n-2) = 2 + T(n-2)$ — ④

Substitute ③ in ④

$T(n) = 2 + 1 + T(n-3) = 3 + T(n-3)$

... $k + T(n-k)$ Stop condition: ~~n=0~~ $n=1$

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

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

$O(n)$

Ex: $T(n) = n + T(n-1) \quad ; \ n > 1$

$\quad\quad\ \ = 1 \quad\quad\quad\quad\quad ; \ n = 1$

$T(n) = n + T(n-1)$ — ①

$T(n-1) = (n-1) + T(n-2)$ — ②

$T(n-2) = (n-2) + T(n-3)$ — ③

...

Subs. ② in ①

$T(n) = n + (n-1) + T(n-2)$ — ④

Subs. ③ in ④

$T(n) = n + (n-1) + (n-2) + T(n-3)$

...

$T(n) = n + (n-1) + (n-2) + (n-3) + \cdots + (n-k) + T(n-(k+1))$

Base condition $n-(k+1) = 1$

$n - k - 1 = 1$

$k = n - 2$

1

2

$T(n) = n + (n-1) + (n-2) + \cdots + (n-(n-2)) + T(n-(n-2+1))$

$= n + (n-1) + (n-2) + \cdots + 2 + 1$

$= \dfrac{n(n+1)}{2}$

$O(n^2)$

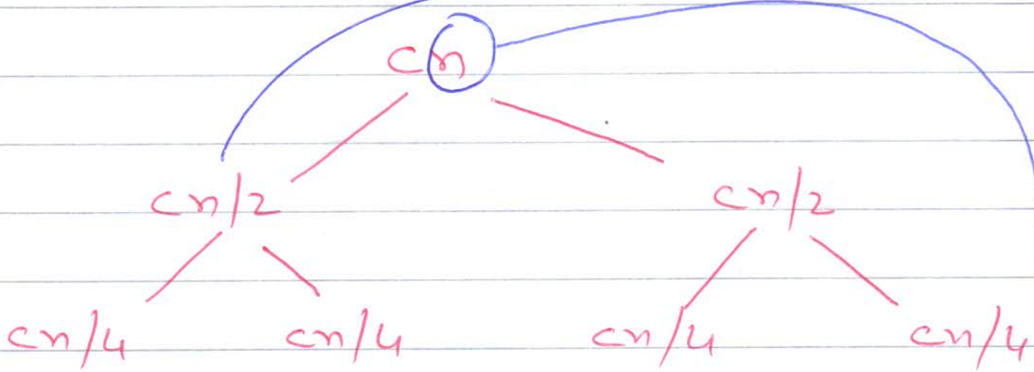
```
fact(int n) {
   if(n=0)
     return 1
   return n*fact(n-1)
}
```

$$T(n) \begin{cases} n * T(n-1) & ; \ n > 0 \\ 1 & ; \ n = 1 \end{cases}$$

Recursion Tree Method

$$T(n) = \begin{cases} 2T(n/2) + cn & ; n > 1 \\ 1 & ; n = 1 \end{cases}$$

$c(n/2)$



Cost

cn

cn

cn

Levels	Nodes
0	$1 = 2^0$
1	$2 = 2^1$
2	$4 = 2^2$
3	$8 = 2^3$
⋮	
$L-1$	2^{L-1}

$$2^{L-1} = n$$

$$L - 1 = \log_2 n$$

$$L = \log_2 n + 1$$

Complexity / Cost = No of levels * Cost per level

$$= (\log_2 n + 1) * cn$$

$$= cn \log_2 n + cn$$

$O(n \log_2 n)$

$$\left. \begin{aligned} T(n) &= 3T(n/3) + cn \\ L &= \log_3 n + 1 \end{aligned} \right\}$$