# COP 3502 10/6/23

Recurrence Relations

Monday: New Problem Analysis

**fact**

```
if (n<=1) return 1;
return n * fact(n-1)
```

let $T(n)$ = run time of fact for input size $n$.

$T(n) = T(n-1) + 1, \; T(1) = 1$

Iteration Technique, Master Thm [plug n chug formula]

$T(n) = \boxed{T(n-1) + 1}$

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

$= \boxed{T(n-2) + 2}$

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

$= \boxed{T(n-3) + 3}$

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

$= T(n-2) + 1$

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

work on side

After k steps, we have

$T(n) = T(n-k) + k, \quad T(1) = 1$

Let $n - k = 1 \rightarrow k = n - 1$

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

$= T(1) + n - 1$

$= 1 + n - 1$

$= n$ $\quad T(n) = n \Rightarrow O(n)$ run-time

# Binary Search worst case

input size $n$

rec left $n/2$

or

rec right $n/2$

let $T(n)$ = run-time binary search of an sorted array of size $n$.

$$T(n) = T\left(\frac{n}{2}\right) + 1, \quad T(1) = 1$$

$$T(n) = \boxed{T\left(\frac{n}{2}\right) + 1}$$

$$= T\left(\frac{n}{4}\right) + 1 + 1$$

$$= \boxed{T\left(\frac{n}{4}\right) + 2}$$

$$= T\left(\frac{n}{8}\right) + 1 + 2$$

$$= \boxed{T\left(\frac{n}{8}\right) + 3}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{\frac{n}{2}}{2}\right) + 1$$

$$= T\left(\frac{n}{4}\right) + 1$$

After $k$ steps we have:

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

Let $\frac{n}{2^k} = 1 \longrightarrow n = 2^k$ and $k = \log_2 n$

$$T(n) = T(1) + \log_2 n$$

$$= 1 + \log_2 n = O(\lg n)$$

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

$c$ = Const

Recurrence for Merge Sort

$$= 2\left[2T\left(\frac{n}{4}\right) + \frac{cn}{2}\right] + cn$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + c \cdot \frac{n}{2}$$

$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + c \cdot \frac{n}{4}$$

$$T(1) = 1$$

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

$$= \boxed{4T\left(\frac{n}{4}\right) + 2cn}$$

$$= 4\left[2T\left(\frac{n}{8}\right) + \frac{cn}{4}\right] + 2cn$$

$$= 8T\left(\frac{n}{8}\right) + cn + 2cn$$

$$= \boxed{8T\left(\frac{n}{8}\right) + 3cn}$$

After $k$ iterations, we have:

$$T(n) = \underline{2^k} T\left(\frac{n}{2^k}\right) + kcn$$

Let $\frac{n}{2^k} = 1 \rightarrow n = 2^k \rightarrow k = \log_2 n$

$$T(n) = nT(1) + (\log_2 n)cn$$

$$= n + cn\log_2 n$$

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

# Towers of Hanoi

**towers(n)**

towers(n-1)
1 move
towers(n-1))

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

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

$$T(n) = \boxed{2T(n-1)+1}$$

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

$$= 2\left[2T(n-2)+1\right]+1$$

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

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

$$= \boxed{4T(n-2)+3}$$

$$= 4\left[2T(n-3)+1\right]+3$$

$$= 8T(n-3) + 4+3$$

$$= \boxed{8T(n-3)+7}$$

$$a^b \times a^c = a^{b+c}$$

$$\underbrace{a \times a \times a \ldots}_{b}$$

$$\underbrace{a \times a \times a \ldots}_{c}$$

a appears b+c times

After k iterations we have:

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

Let $n-k=1 \rightarrow k = n-1$

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

$$= \underbrace{2^{n-1} + 2^{n-1}} - 1$$

$$= 2^1 \times 2^{n-1} - 1$$

$$= \boxed{2^n - 1}$$

run time

$O(2^n)$

$T(n) = \boxed{3T\left(\frac{n}{2}\right) + n}$ , $T(1) = 1$

$T\left(\frac{n}{2}\right) = 3T\left(\frac{n}{4}\right) + \frac{n}{2}$

$T\left(\frac{n}{4}\right) = 3T\left(\frac{n}{8}\right) + \frac{n}{4}$

$= 3\left[3T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n$

$= \boxed{9T\left(\frac{n}{4}\right) + \frac{3n}{2} + n}$

$= 9\left[3T\left(\frac{n}{8}\right) + \frac{n}{4}\right] + \left(n + \frac{3}{2}n\right)$

$= 27T\left(\frac{n}{8}\right) + \frac{9n}{4} + n + \frac{3}{2}n$

$= \boxed{27T\left(\frac{n}{8}\right) + \left(n + \frac{3}{2}n + \frac{9}{4}n\right)}$

After $k$ steps we have

$$T(n) = 3^k T\left(\frac{n}{2^k}\right) + n \sum_{i=0}^{k-1} \left(\frac{3}{2}\right)^i$$

Let $\frac{n}{2^k} = 1 \rightarrow n = 2^k \Rightarrow k = \log_2 n$

$$T(n) = 3^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{3}{2}\right)^i$$

$$= n^{\log_2 3} + n \left[\frac{\left(\frac{3}{2}\right)^{\log_2 n} - 1}{\frac{3}{2} - 1}\right]$$

$$= n^{\log_2 3} + 2n\left[\cancel{\log}\ n^{\log_2\left(\frac{3}{2}\right)} - 1\right]$$

$$= n^{\log_2 3} + 2n\left[n^{\log_2 3 - \log_2 2} - 1\right]$$

$$= n^{\log_2 3} + 2n\left[\frac{n^{\log_2 3}}{n} - 1\right]$$

$$= n^{\log_2 3} + 2n^{\log_2 3} - 2n$$

$$= O(n^{\log_2 3})$$

---

Master Thm

$$T(n) = A\,T\left(\frac{n}{B}\right) + O(n^k)$$

if $B^k > A$, then $O(n^k)$

if $B^k = A$, then $O(n^k \lg n)$

if $B^k < A$, then $O(n^{\log_B A})$

$T(n) = T\left(\frac{n}{2}\right) + 1$, $A = 1$, $\underline{\underline{B = 2}}$, $k = 0$, $B^k = 1$, $A = 1$ ✓

$B^k = A \Rightarrow O(\lg n)$

$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$ $A = 2, B = 2, k = 1, B^k = A = 2$

$B^k = A \Rightarrow O(n \lg n)$

$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$ $A = 3, B = 2, k = 1, B^k = 2, A = 3$

$B^k < A \rightarrow O(n^{\log_2 3})$

$T(n) = 3T\left(\frac{n}{2}\right) + O(n^2)$ $A = 3, B = 2, k = 2, B^k = 4, A = 3$

$B^k > A \Rightarrow O(n^2)$