COP 3502 10/5/23

(1) Recurrence Relations - Iteration Technique
- Master Thm

(2) Alg Analysis - New Problem Analysis

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

Let $T(n)$ be the run-time of fact(n)

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

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

$= T(n-2) + 1$

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

$T(1) = 1$

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

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

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

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

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

$T(n) = n$

After k iterations, we have

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

since $T(1) = 1$
let $n-k = 1$
$k = n-1$

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

$= T(1) + n - 1$

$= 1 + n - 1$

$= n$

$O(n)$

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

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

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

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

$$= 4(2T(n-3) + 1) + 3$$

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

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

towers(n)

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

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

After k iterations we have:

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

Let $n-k = 0 \Rightarrow k = n$.

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

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

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

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

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

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

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

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

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

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

After k iterations we have

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

We know $T(1)$. Let $\frac{n}{2^k} = 1 \Longrightarrow n = 2^k$ and $k = \log_2 n$

$$T(n) = nT\left(\frac{n}{n}\right) + (\log_2 n) \cdot cn$$

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

$$= n + cn\log_2 n$$

$$= \boxed{O(n \lg n)}$$

Binary Search Recurrence

$$T(n) = T\left(\tfrac{n}{2}\right) + \cancel{O(1)}\, c \quad , T(1)=1$$

$$= T\left(\tfrac{n}{4}\right) + c + c$$

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

$$= T\left(\tfrac{n}{8}\right) + c + 2c$$

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

After k iterations, we have

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

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

$$T(n) = T\left(\tfrac{n}{n}\right) + (\log_2 n)c$$

$$= 1 + c\log_2 n$$

$$= O(\lg n)$$

aside

$$(x+y)^2 = x^2 + 2xy + y^2$$

$$\Big(\underbrace{\overset{x}{(n^2-2n)}}_{x} + \overset{y}{4}\Big)^2 = (n^2-2n)^2 + 8(n^2-2n) + 16$$

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

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

$$= 9T\left(\frac{n}{4}\right) + \left(\frac{3}{4}n^2 + n^2\right) \qquad T\left(\frac{n}{4}\right) = 3T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2 = 3T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$= 9\left[3T\left(\frac{n}{8}\right) + \frac{n^2}{16}\right] + \left(\frac{3}{4}n^2 + n^2\right)$$

$$= 27T\left(\frac{n}{8}\right) + \left(n^2 + \frac{3}{4}n^2 + \frac{9}{16}n^2\right)$$

After k iterations we have

$$= 3^k T\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} n^2 \left(\frac{3}{4}\right)^i$$

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

$$T(n) = 3^{\log_2 n} T(1) + n^2 \sum_{i=0}^{\log_2 n} \left(\frac{3}{4}\right)^i$$

$$\leq n^{\log_2 3} + n^2 \cdot \frac{1}{\left(1 - \frac{3}{4}\right)}$$

$$\leq n^{\log_2 3} + 4n^2$$

$$= O(n^2)$$

Karatsuba's Alg.

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

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

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

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

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

After k iterations,

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

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

$$T(n) = 3^{\log_2 n} T(1) + n \cdot \frac{\left(\frac{3}{2}\right)^k - 1}{\left(\frac{3}{2}\right) - 1}$$

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

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

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

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

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

Many recursive algs have a run-time that can be solved via recurrence of this form:

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

rec calls

size of each rec call

extra work

Master Thm

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

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

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

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

Case 2 $\quad B^k = A \rightarrow O(n \lg n)$

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

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

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

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

Sec Q

Fall 2023 C1

2^n strings

for each run an $O(n)$ loop to count consecutive Bs.

total time $= O(n \cdot 2^n)$

Sum 2023 C1

Let $T(n) =$ amt mem created by ...

Can use Master

$A = 1$ $k = 1$

$B = 2$ $\longrightarrow O(n)$

$$
\begin{aligned}
T(n) &= 1 + n + T\left(\tfrac{n}{2}\right) \\
&= T\left(\tfrac{n}{2}\right) + O(n) \\
&= T\left(\tfrac{n}{4}\right) + c\tfrac{n}{2} + cn \\
&= T\left(\tfrac{n}{8}\right) + c \cdot \tfrac{n}{4} + c\tfrac{n}{2} + cn \\
&= T\left(\tfrac{n}{2^k}\right) + cn \sum_{i=0}^{k-1} \left(\tfrac{1}{2}\right)^i, \quad T(1) = 1 \\
&= T(1) + cn \sum_{i=0}^{\log_2 n - 1} \left(\tfrac{1}{2}\right)^i \\
&\le 1 + cn \sum_{i=0}^{\infty} \left(\tfrac{1}{2}\right)^i \\
&= 1 + cn \left(\frac{1}{1 - \frac{1}{2}}\right) = \boxed{O(n)}
\end{aligned}
$$

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

$n = 2^k$

$k = \log_2 n$

Sum 2022 C1

Code

2 rec calls size $\frac{n}{2}$

for loops $O(n)$

$T(n) = 2T(\frac{n}{2}) + O(n)$ $A=2$, $B=2$, $k=1$ $\rightarrow O(n \lg n)$

$B^k = A$ case 2

Spr 2022 C1

best $O(1)$ can return in loop 1^{st} time

worst $O(\lg n)$ - start at $\frac{n}{2}$ and repeatedly divide by 2.

Fall 2020 C1

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

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

$= \boxed{T(n-3) + (n-2) + (n-1) + n}$

...

$= T(1) + \sum_{i=2}^{n} i = \frac{n(n+1)}{2} = O(n^2)$