# Lecture 4

Lecture 4

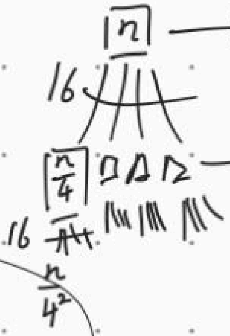
LAST Time

recurrence tree method for solving closed form of recurrences.

$T(n) = 16T(\frac{n}{4}) + n^3$

$16T(\frac{n}{4}) \begin{cases} \text{for } (i=1 \text{ to } 10)\{ \\ \text{Foo } (A(1 \dots \frac{n}{4})) \end{cases}$

```
for (j=1 to n)
    for (i=j to n)
        for (k=1 to n)
            print hi;
```




$\rightarrow n^3$

$\rightarrow 16(\frac{n}{4})^3$

$16^2(\frac{n}{4^2})^3$

work $\boxed{16^i(\frac{n}{4^i})^3} = 16^i \frac{n^3}{(4^i)^3}$

$= \frac{16^i}{(4^3)^i} n^3$

$= (\frac{16}{4^3})^i n^3$

$= (\frac{1}{4})^i n^3$

$r < 1$

depth $\log_4 n$

$\sum_{i=0}^{\log_4 n} (\frac{1}{4})^i n^3$

$n^3 \sum_{i=0}^{\log_4 n} (\frac{1}{4})^i \le n^3 \sum_{i=0}^{\infty} (\frac{1}{4})^i$

$\frac{r^{n+1}-1}{r-1}$

$\sum i = \frac{n(n+1)}{2}$ work at level

$\sum$ constant

depth of levels

$\sum r^i \begin{cases} |r| < 1 \rightarrow \frac{1}{1-r} \\ |r| > 1 \quad \boxed{\frac{r^{n+1}-1}{r-1}} \end{cases}$

$\sum$ work
depth

$\frac{n}{4^i} = 1$

$n = 4^i$

$\boxed{\log_4 n = i}$

$n^3 \sum_{i=0}^{\log_4 n} (\frac{1}{4})^i \le n^3 \sum_{i=0}^{\infty} (\frac{1}{4})^i$

$n^3 \frac{1}{1 - 1/4} = n^3 (\frac{4}{3}) = O(n^3)$

## Substitution

$\underset{k-1}{1} \; \underset{k}{2} \; 3 \; \ldots\ldots$

1. guess
2. verify via induction using def of Big-O
3. ↳ Solve for constants, $c, n_0$

$T(1) = 1 \le c \cdot 1$

$$T(n) = T\left(\frac{n}{6}\right) + T\left(\frac{2n}{6}\right) + O(n)$$

guess $T(n) = O(n)$

$\exists\, c, n_0 \quad T(n) \le c \cdot n$

I.H. **Assume** $T(n) = O(n)$ for all $n \le k-1$.

$T(n) \le cn \quad n \le k-1$

$\boxed{n=k}$? $\quad T(k) = T\left(\frac{k}{6}\right) + T\left(\frac{2k}{6}\right) + O(k)$

$\le$ by IH $\le$ $\quad \le$ ; def of B-O

$$\underbrace{c\left(\frac{k}{6}\right) + c\frac{2k}{6} + ak}\ \boxed{\le}\ ck$$

$$\frac{3}{6}ck + ak \le ck$$

$$\frac{1}{2}c + a \le c$$

$$a \le \frac{1}{2}c$$

$$\boxed{2a \le c}$$

$T(n) = 2T\left(\frac{n}{2}\right) + O(n) \qquad n\log n$

guess $T(n) = O(n)$

Assume $T(n) \le cn$ for all $n \le k-1$

$\boxed{n=k}$? $\quad T(k) = 2T\left(\frac{k}{2}\right) + k\ \boxed{\le}\ ck$

$\Downarrow$

$\exists c$? $\quad 2c\frac{k}{2} + k \le ck$

$\boxed{ck} + k \le \boxed{ck}$

---

$T(n) = 2T\left(\frac{n}{2}\right) + n \qquad T(n) = O(n\log n)$

$T(n) = O(n\log n)$
$\le cn\log n$

step1 state inductive hypothesis

Assume $T(n) \leq cn\log n$ for all $n \leq k-1$

1 $k-1$

step2 solve for $c, n_0$ when $n=k$

$\boxed{n=k}$? $T(k) = 2T(\frac{k}{2}) + k \boxed{\leq} ck\log k$

$\leq$

$2c\frac{k}{2}\log(\frac{k}{2}) + k$

$2ck[\log k - \log_2 2] + k \leq ck\log k$

$T(1)=1$ $ck\log k - ck + k \leq ck\log k$

$\exists\, c=1$ $-ck + k \leq 0$

$\boxed{n_0=1}$ $1k \leq ck\,1$

$\boxed{1 \leq c}$

---

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

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

step1: IH $T(n) \leq cn^2$ for all $n \leq k1$ ✓

step2: solve for $c, n_0$ when $n=k$

$\boxed{n=k}$? $T(k) = 2T(\frac{k}{2}) + k \boxed{\leq} ck^2$

$\leq$ IH

$2c(\frac{k}{2})^2 + k \leq ck^2$

$\exists \boxed{c=2}$ $\frac{1ck^2}{2} + k \leq ck^2$

$n_0=1$

$\frac{k}{k} \leq \frac{1}{2}ck^2/k$

$1 \leq \frac{1}{2}ck$

$$T(n) = T\left(\frac{n}{7}\right) + T\left(\frac{5n}{7}\right) + O(n)$$

guess $T(n) = O(n)$

step 1: Assume $T(n) \le cn$ $\boxed{\text{for all}}$ $\boxed{n \le k-1}$

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

step 2: find $n_0$, $c$, for $n=k$

$\to n$

$\to n$

$n=k$ $\quad T(k) = T\left(\frac{k}{7}\right) + T\left(\frac{5k}{7}\right) + O(k) \le ck$

linear-selection

$\Downarrow \le IH \quad \Downarrow \le IH \quad \le$

$$c\frac{k}{7} + c\frac{5k}{7} + ak \boxed{\le} ck$$

$$6\frac{ck}{7} + ak \le ck$$

$\exists$ $\boxed{c = 7a}$ $n_0 = 1$

$$a\cancel{k} \le \frac{1}{7}c\cancel{k} \cdot 7$$

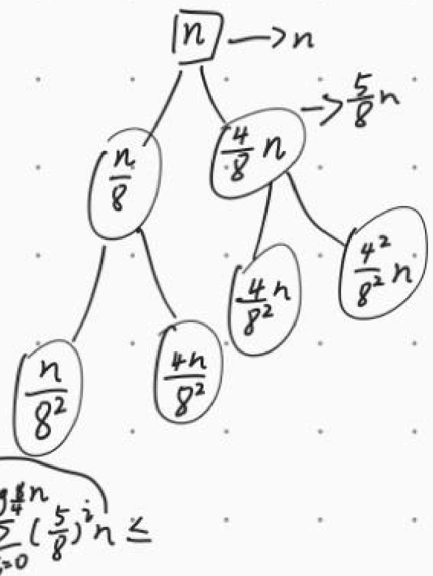
$$7a \le c$$

$\boxed{O(n^3) \le an^3}$

practice

① $T(n) = 4T\left(\frac{n}{2}\right) + O(n)$ $\quad$ guess $T(n) = O(n\log n)$

② $T(n) = T\left(\frac{1}{8}n\right) + T\left(\frac{4}{8}n\right) + O(n)$




$$\frac{4}{8}\,\frac{n}{8} = \frac{4n}{8^2}$$

$\frac{4}{8}n\left(\frac{1}{8}\right)$ AND $\frac{4}{8}n\left(\frac{4}{8}\right)$

$$\frac{5^2}{8^2}n$$

work $\frac{5^i}{8^i}n$

Depth: $\left(\frac{4}{8}\right)^i n = 1$

$$n = \left(\frac{8}{4}\right)^i$$

$$\boxed{\log_{\frac{8}{4}} n = i}$$

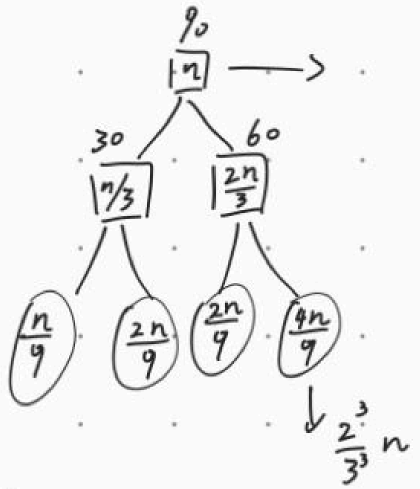
$$\sum_{i=0}^{\log_{\frac{8}{4}} n} \left(\frac{5}{8}\right)^i n \le$$

$$n\sum_{i=0}^{\infty}\left(\frac{5}{8}\right)^i \le n\frac{1}{1-5/8} = O(n)$$

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

$n$

$n = 90$




$$\frac{n}{3} + \frac{2n}{3} = n$$

$$\frac{n}{9} + \frac{2n}{9} + \frac{2n}{9} + \frac{4n}{9} = \frac{9n}{9} = n$$

$\boxed{\text{work at level: } n}$

$$\left(\frac{2}{3}\right)^i n = 1$$

$n = \left(\frac{3}{2}\right)^i$ $\boxed{\log_{3/2} n = i}$ ← depth

$$\sum_{i=0}^{\log_{3/2} n} n = n \sum_{i=0}^{\log_{3/2}} 1 = n(\log_{3/2} n + 1) = O(n \log n)$$

- Recurrence Tree method
- Substution
- master Theorem method

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

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

$A = 4 \quad B = 2 \quad D = 2$

$\log_B A$ vs $D$

$\log_B A > D \Rightarrow O(n^{\log_B A})$

if $\log_B A = D \Rightarrow O(n^{\log_B A} \log n)$

$\log_B A < D \Rightarrow O(n^d)$

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

$A=9 \quad B=3 \quad D=1$

$\log_3 9$ 2 vs. 1 $O(n^2)$