Divide and Conquer Algorithms

Tuesday, October 13, 2020 5:00 PM

Reminder: HW5 is due this Sunday.

Example:

from L13

⑤ $T(n) = 6T(n-1) + 1$

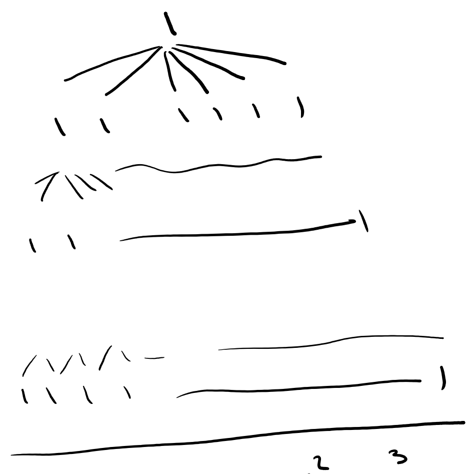
⑥ $T(n) = T(\frac{n}{2}) + 1$

⑤ $T(n) = 6T(n-1) + 1 \longrightarrow T(\boxed{n-1}) = 6T(\boxed{n-1} - 1) + 1$

step	size
0	$n - 0$
1	$n - 1$
2	$n - 2$
⋮	
k	$1 = n - k$

$k = n - 1$

tree



$T(n) = 1 + 6 + 6^2 + 6^3 + \cdots + 6^k$

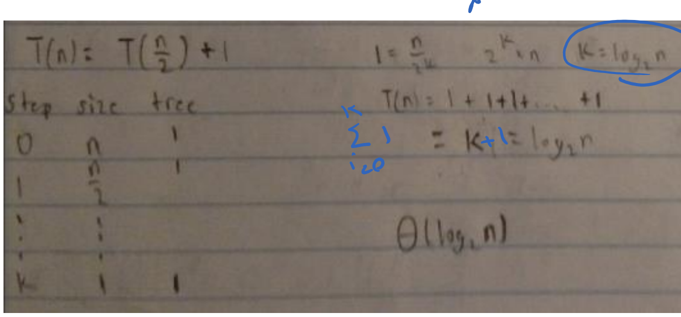
$= \sum_{i=0}^{k} 6^i = \frac{6^{k+1} - 1}{6 - 1} \simeq 6^{k+1} = 6^{n-1+1}$

$\boxed{= O(6^n)}$

Binary-Search

⑥ $T(n) = T(\frac{n}{2}) + 1 \longrightarrow T(\boxed{n/2}) = T\left(\frac{\boxed{n/2}}{2}\right) + 1$

Math



$T(n) = T(\frac{n}{2}) + 1$ $1 = \frac{n}{2^k}$ $2^k = n$ $k = \log_2 n$

step	size	tree
0	n	1
1	$\frac{n}{2}$	1
⋮	⋮	
k	1	1

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

$\sum_{i=0}^{k} 1 = k + 1 = \log_2 n$

$\Theta(\log_2 n)$

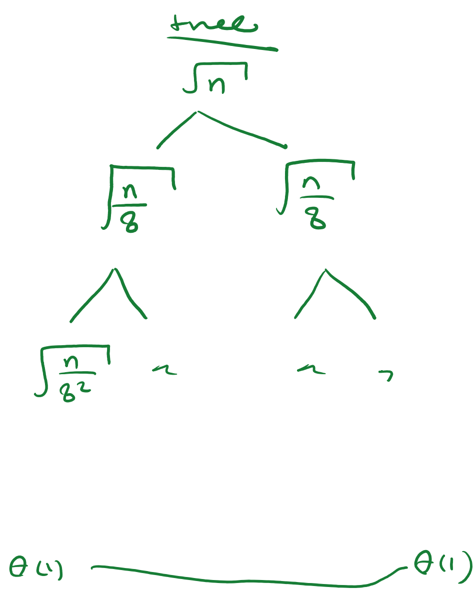
⑦ $T(n) = 2T\left(\frac{n}{8}\right) + \sqrt{n}$

⑧ $T(n) = T(n-2) + 1$

⑨ $T(n) = T(n-1) + \log n$

⑦ $T(n) = 2T\left(\frac{n}{8}\right) + \sqrt{n}$

step	size
0	$\frac{n}{8^0}$
1	$\frac{n}{8^1}$
2	$\frac{n}{8^2}$
⋮	
K	$1 = \frac{n}{8^K}$



$n = 8^K$

$\boxed{K = \log_8 n}$

$$T(n) = \sqrt{n} + 2\sqrt{\frac{n}{8}} + 2^2\sqrt{\frac{n}{8^2}} + \sim + 2^{K-1}\sqrt{\frac{n}{8^{K-1}}} + 2^K$$

$$= \sum_{i=0}^{K-1} 2^i \sqrt{\frac{n}{8^i}} + 2^K$$

$$= \sqrt{n} \sum \frac{2^i}{(\sqrt{8})^i} + 2^K = \sqrt{n} \sum \left(\frac{2^i}{2^{1.5}}\right) + 2^K$$

$$= \sqrt{n} \sum_{i=0}^{K-1} \left(\frac{1}{\sqrt{2}}\right)^i + 2^K$$

$$= \sqrt{n} \frac{\cancel{\left(\frac{1}{\sqrt{2}}\right)^K} \to 0 \; - 1}{\frac{1}{\sqrt{2}} - 1} + 2^K$$

$\simeq \sqrt{n} + 2^k$

$= \sqrt{n} + 2^{\log_8 n}$ $= \sqrt{n} + n^{\log_8 2}$

$= n^{1/2} + n^{\frac{\log 2}{\log 8}}$

$= n^{1/2} + n^{1/3}$

$\boxed{= \Theta(n^{1/2})}$

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

step	size	tree
0	n	1
1	$n-2$	1
2	$n-4$	1
3	$n-6$	1
k	$1 = n - 2k$	1

$n - 1 = 2k$

$\frac{n-1}{2} = k$

$T(n) = 1 + 1 + 1 \text{ ——— } + 1$

$= \sum_{i=0}^{k} 1 = k + 1$

$= \frac{n-1}{2} + 1$

$\boxed{= \Theta(n)}$

(9) $T(n) = T(n-1) + \log n$ ← note: $f(n) = \log n$ $\quad f(n-1) = \log(n-1)$

step	size	tree

step	size	tree
0	n	$\log n$
1	$n-1$	$\log(n-1)$
2	$n-2$	$\log(n-2)$
$\vdots$	$\vdots$	$\vdots$
K	$1 = n-k$	$\Theta(1)$
	$k = n-1$	

$$T(n) = \underbrace{\log n}_{\text{step } 0} + \underbrace{\log(n-1)}_{\text{step } 1} + \sim + \underbrace{\log 1}_{\text{step } k}$$

$$= \sum_{i=1}^{n} \log i = \Theta\left(\int_1^n \log x \, dx\right) = \boxed{\Theta(n \log n)}$$