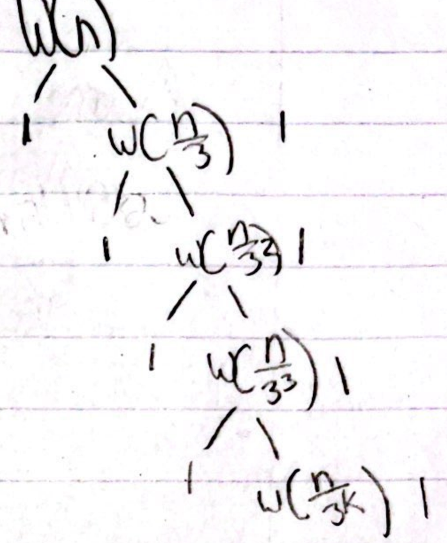
Assignment 2

1. $W(n) = 2W\left(\frac{n}{3}\right) + 1$

$$W(n) = \begin{cases} 1 & n = 1 \\ 2W\left(\frac{n}{3}\right) + 1 & n > 1 \end{cases}$$



$\frac{n}{3^k} = 1$

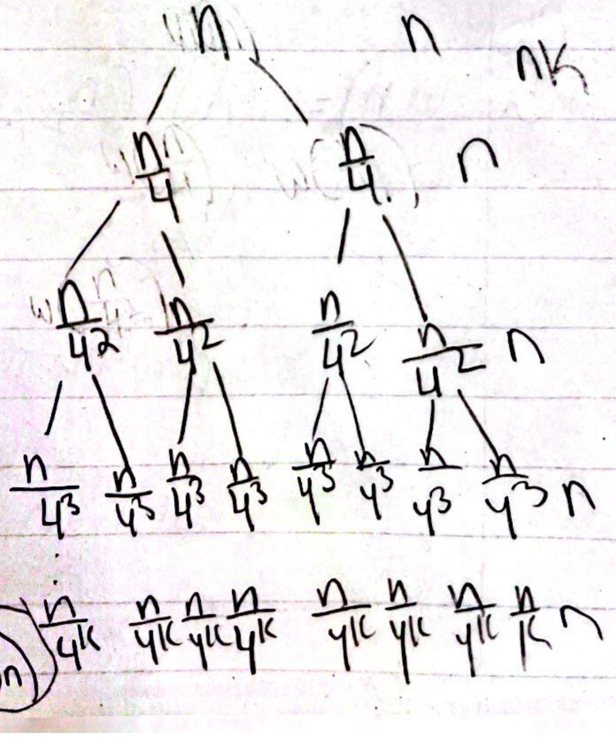
$n = 3^k$

$k = \log_3 n$

$O(\log n)$

2. $W(n) = 5W\left(\frac{n}{4}\right) + n$

$$W(n) \begin{cases} 1 & n = 1 \\ 5W\left(\frac{n}{4}\right) + n & n > 1 \end{cases}$$



Assume: $\frac{n}{4^k} = 1$

$n = 4^k$

$k = \log n$

$nk = n \log n$

So $W(n) = 5W\left(\frac{n}{4}\right) + n = O(n \log n)$

3. $w(n) = 7w\left(\frac{n}{7}\right) + n$

Asume: $\frac{n}{7^k} = 1$

$n^k = \log n$

$k = n \log n$

$O(n \log n)$

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

Asume: $\frac{n^2}{3^k} = 1$

$n^2 k = \log n$

$k = n^2 \log n$

$O(n^2 \log n)$

5. $w(n) = 8W\left(\frac{n}{2}\right) + n^3$

$n^{\log_b a} = n^{\log_2 8} = n^3$

$f(n) = n^3$ equal to n^3

For: $O(f(n) \cdot \log^k(n))$

if $k = 0$:

$O(n^3 \cdot \cancel{\log^0(n)})$

so $O(n^3)$

6. $49w\left(\frac{n}{25}\right) + n^{3/2}\log n$

$$\log_{25} 49 = \frac{\log 7}{\log 5} = 1.2$$

$$3/2 = 1.5$$

$$1.2 < 3.2$$

So $O(n^{3/2} \log n)$

7. $w(n) = w(n-1) + 2$

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

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

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

$w(n-2) = w(n-3) + 2$

Substitute $w(n-1)$

$w(n) = [w(n-2) + 2] + 2$

$w(n) = w(n-2) + 4$

$w(n) = [w(n-3) + 4] + 2$

$w(n) = w(n-3) + 6$

$\vdots$

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

Assume $n-k=0$

$n=k$

$w(n) = w(n-n) + n$

$w(n) = w(0) + n$

$w(n) = 1 + n$

$O(n)$

8. $W(n) = W(n-1) + n^c$, with $c \geq 1$

$w(n-1) = w(n-1-1) + n-1^c$

$w(n-1) = w(n-2) + n-1^c$

$w(n) = [w(n-2) + n-1^c] + n^c$

$w(n) = w(n-2) + n-1^c + n^c$

$w(n-2) = w(n-2-1) + n-2^c$

$w(n-2) = w(n-3) + n-2^c$

$w(n) = [w(n-3) + n-2^c] + n-1^c + n^c$

$w(n) = w(n-3) + n-2^c + n-1^c + n$

⋮

$w(n) = (n-k) + n^c + (n-1)^c + (n-2)^c + \dots + (n-k+1)^c$

Assume $n-k=1$

$n^c + (n-1)^c + (n-2)^c \dots$

$O(n^{c+1})$

9. $w(n) = w(\sqrt{n}) + 1$

$w(n) = w(\sqrt{n}) + 1$

$w(n) = w(n^{1/2}) + 1$

$w(n) = w(n^{1/2^2}) + 2$

$w(n) = w(n^{1/2^3}) + 3$

$w(n) = w(n^{1/2^k}) + k$

Assume $n = 2^m$

$w(2^m) = w(2^{\frac{m}{2^k}}) + k$

Assume $w(2^{\frac{m}{2^k}}) = w(2^1)$

$\therefore \frac{m}{2^k} = 1$

$m = 2^k$ & $k = \log_2 m$

$n = 2^m$ & $m = \log_2 n$

$k = \log \log_2 n$

$O(\log \log_2 n)$

Scanned with CamScanner

2.

$a = 5$, $b = 2$

A: 5 problems (size of subproblem)

- cut in half: $\frac{n}{2}$
- add them up linearly: $O(n)$

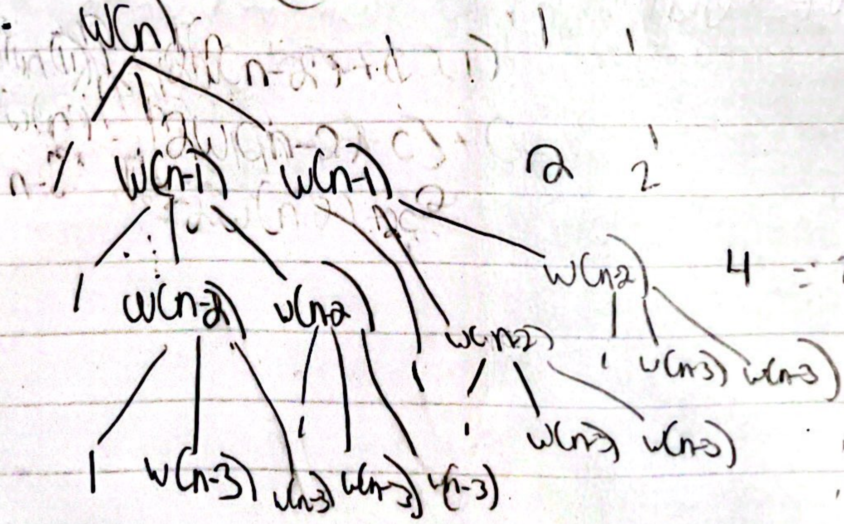
$$W(n) = 5W\left(\frac{n}{2}\right) + O(n)$$

$$n^{\log_b a} = n^{\log_2 5}$$

$$O\left(n^{\log_2 5}\right) = O(n^{2.3})$$

B:

$$W(n) = 2W(n-1) + O(1)$$



1

2

$4 = 2^2$

$8 = 2^3$

2^k

$$1 + 2 + 2^2 + 2^3 + \dots + 2^k = \frac{1(2^{k+1} - 1)}{2-1} = 2^{k+1} - 1$$

Assume $n - k = 0$

$n = k$

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

$$= O(2^n)$$

C. $W(n) = 9w\left(\frac{n}{3}\right) + n^2$

$a = 9 \quad b = 3 \quad f(n) = n^2$

$\log_b a \quad n^{\log_3 9} = 2$

$f(n) = n^2$ is equal to $n^{\log_3 9} = n^2$

So its balanced

$O(n^2 \log n)$

- I think Algorithm C b/c even though there's a log, the n^2 makes it faster.

Algorithm A also is faster if there is a smaller n

but $O(n^2 \log n)$ is faster for larger n