jongminl-311-hw2

jongminl Lee

February 2024

1

(a) Since it asks to make table

0	14
1	
2	44
3	4, 15
4	19, 41
5	23
6	1
7	9, 42
8	24
9	17
10	10, 43
11	21

2

(a) For $T(n) = 7T(\frac{n}{2}) + 1$ a = 7, b = 2, f(n) = 1 Since $g(n) = n^{log_2 7}$ which is bigger than f(n) that means g(n) becomes runtime. Which become $\theta(n^{log_2 7})$

(b) For induction to prove we first need base case. $T(1) = \frac{1}{6}(7(1^{log_2 7}) - 1)$ Since 1^n is just 1. $\frac{1}{6}(7 * 1 - 1) = \frac{1}{6}(6) = 1$ so we know our base case is true. next we can prove next condition which is T(n+1). $T(n+1) = 7 * T(\frac{n+1}{2}) + 1$ $= 7(\frac{1}{6}(7(\frac{n+1}{2} - 1)) + 1 = 7 * (\frac{1}{6}(7(\frac{n^{log_2 7}+1}{2^{log_2 7}} - 1)) + 1 = 7(\frac{1}{6}(7(\frac{n^{log_2 7}+1}{7})) + 1 =$ $7(\frac{1}{6}(n^{log2_7} + 1 - 1) + 1 = 7(\frac{n^{log_2 7}}{6}) + 1 = \frac{7n^{log_2 7}}{6} + \frac{6}{6} = \frac{7n^{log_2 7}+6}{6}$

3

(a)

0	$\frac{n}{2^0}$	$2^0 n$
1	$\frac{n}{2^1}$	$2^1 n$
2	$\frac{n}{2^2}$	$2^2 n$
3	$\frac{n}{2^3}$	$2^3 n$
k	$\frac{n}{2^k}$	$2^k n$

Since it increased by power of 2, and also T(n) = 1 when n

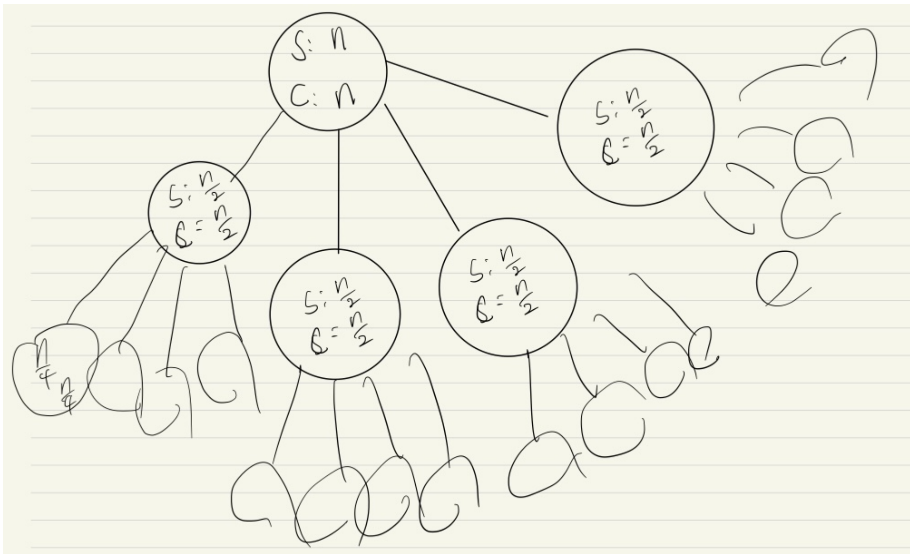

Figure 1: (a)

¡ 2 we can see that $\frac{n}{2^k} = 1 = n = 2^k = k = log_2 n$ $2^l og_2 n * n = 2^l og_2 n = n = n * n = n^2$ For the conclusion, O(n^2)

(b)

0	$\frac{n}{2^0}$	$nlogn$
1	$\frac{n}{2^1}$	$\frac{n}{2} log \frac{n}{2}$
2	$\frac{n}{2^2}$	$\frac{n}{2^2} log \frac{n}{2^2}$
3	$\frac{n}{2^3}$	$\frac{n}{2^3} log \frac{n}{2^3}$
k	$\frac{n}{2^k}$	$\frac{n}{2^k} log \frac{n}{2^k}$

As you can see table, it's cost grows by $\frac{n}{2^k}$

(c)

0	n	n
1	$\frac{n}{3}$	$\frac{n}{3}$
2	$\frac{n}{3^2}$	$\frac{n}{3^2}$
3	$\frac{n}{3^3}$	$\frac{n}{3^3}$
k	$\frac{n}{3^k}$	$\frac{n}{3^k}$

As you can see the table and figure, $\frac{n}{3^k} = 1 = n = 3^k = k = log_3 n$ Then $\frac{n}{3^k} = \frac{n}{3^{log_3 n}}$ Since $3^{log_3 n} = n$ So that $\frac{n}{3^{log_3 n}} = \frac{n}{n} = 1$ For conclusion, O(1)

(d)

0	$3^0 n$	$n^2.5$
1	$3^1 n$	$n^2.5^2$
2	$3^2 n$	$n^2.5^3$
3	$3^3 n$	$n^2.5^4$
k	$3^k n$	$n^2.5^k$

From the table and figure, $3^k n = 1 = 3^k = n = k = log_3 n$

Which is $n^{2.5} = (n^{2.5^{log_3 n}}) = n^{n^{log_3 2.5}}$ For the conclusion, O($n^{n^{log_3 2.5}}$)

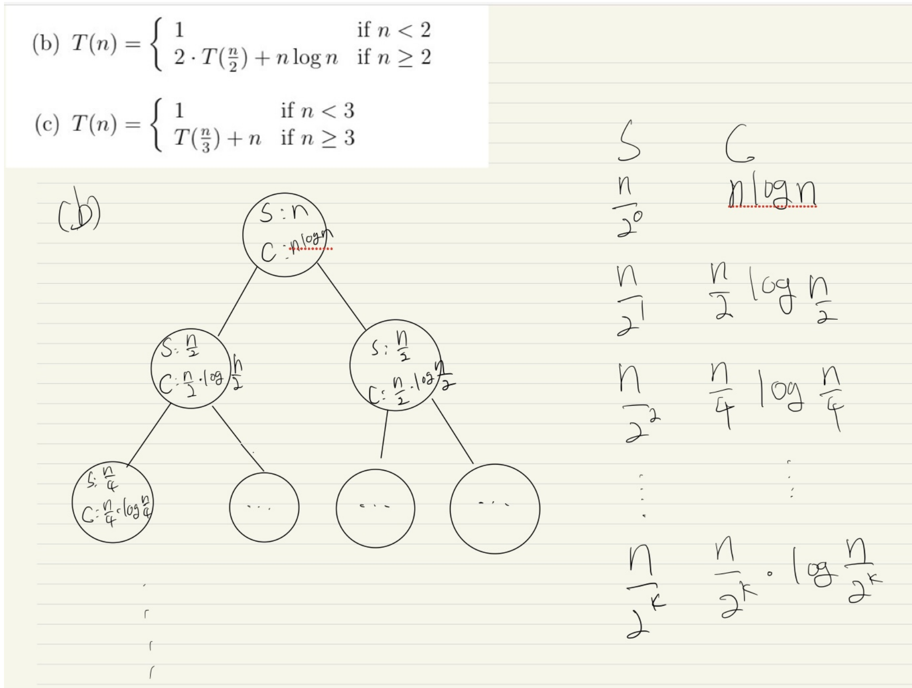

Figure 2: (b)

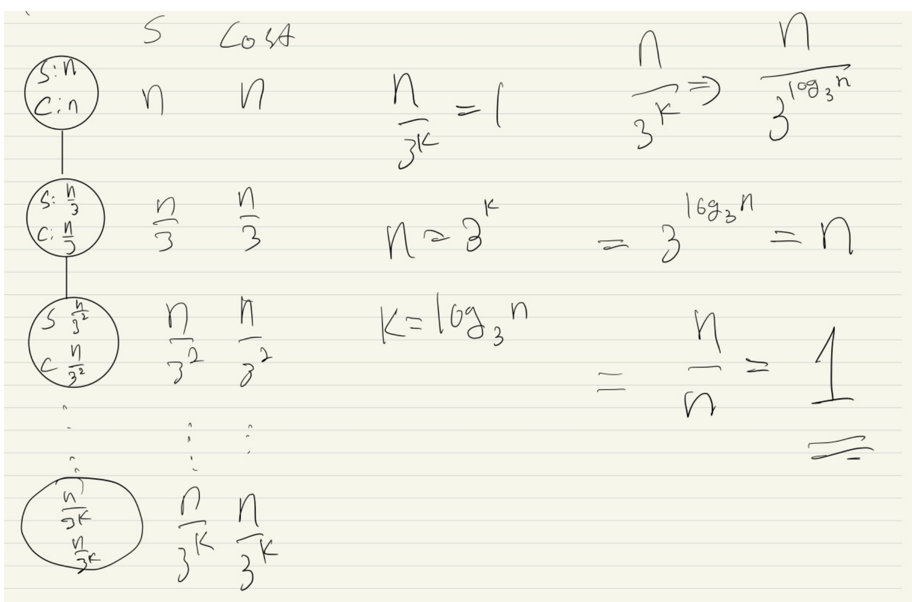

Figure 3: (c)

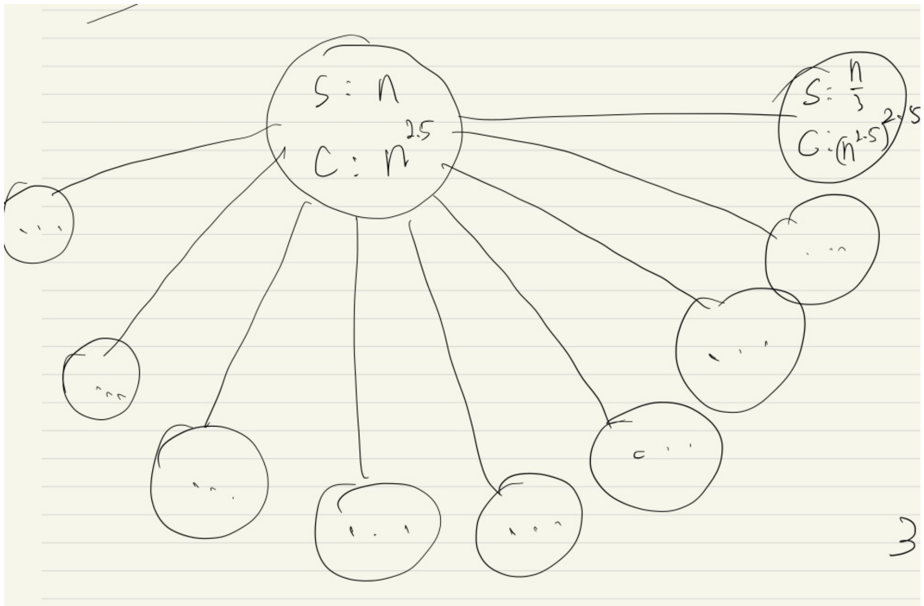

Figure 4: (d)

(e)

0	$n-2$	n^2
1	$n-2*2$	n^{2^2}
2	$n-2*3$	n^{2^3}
3	$n-2*4$	n^{2^4}
k	$n-2*k$	n^{2^k}

As from table and figure, $n-2k=1$ which is $n-1=2k$ then $k=\frac{n-1}{2}$. Thus, $n^{2^{\frac{n-1}{2}}}$ which become O($n^{2^{\frac{n-1}{2}}}$)

4

(a) $T(n)=4T(\frac{n}{2})+n$ From this, a = 4, b = 2, f(n) = n since g(n) = $\theta(n^{log_b a})$ = $\theta(n^{log_2 4})$ = $\theta(n^2)$ Since f(n) = n, g(n) = n^2 which means g(n) ¿ f(n) When g(n) is bigger then f(n), g(n) becomes runtime. For the conclusion O(n^2)

(b) $T(n)=2T(\frac{n}{2})+nlogn$ From this equation, a = 2, b = 2, f(n) = nlogn g(n) = $n^{log_a b}=n^{log_2 2}=n^1$ f(n) ¿ g(n) = f(n) = $\theta(fn)=a*f(n)\leq c*f(n)$ only when c ¡ 1. = $2*f(\frac{n}{2})\leq c*f(n)=2*\frac{n}{2}*log\frac{n}{2}\leq c*nlogn=\frac{nlog\frac{n}{2}}{nlogn}\leq C=\frac{logn-log2}{logn}\leq C$ but since it can only applied to C ¡ 1 it cannot be applied.

(c) $T(n)=3T(\frac{n}{3})+n$ From this equation, a = 3, b = 3, f(n) = n g(n) = $n^{log_b a}=n^{log_3 3}=n^1$ Since f(n) = g(n), when g(n) is equal to f(n) then runtime become g(n) * logn, which is nlogn So the runtime is $\theta(nlogn)$

(d) T(n) = $2*T(\frac{n}{4})+\sqrt{n}$ From this equation, a = 2, b = 4, f(n) = $\sqrt{n}$ g(n) = $n^{log_b a}=n^{log_4 2}=n^{\frac{1}{2}}=\sqrt{n}$ Since f(n) = g(n), runtime is $\theta(\sqrt{n)}$

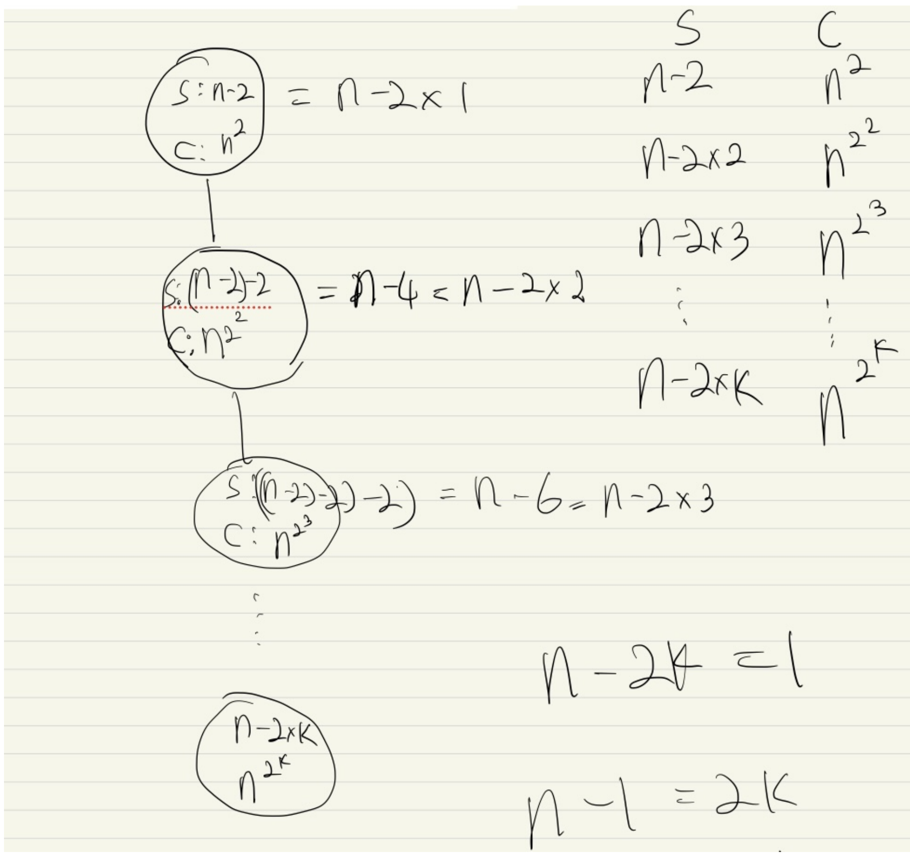
S
C
s: n-2
c: n^2
= n-2x1
n-2
n^2
n-2x2
n^2^2
n-2x3
n^2^3
s: (n-2)-2
c: n^2
= n-4 = n-2x2
n-2xk
n^2^k
s: ((n-2)-2)-2)
c: n^2^3
= n-6 = n-2x3
n-2k = 1
n-2xk
n^2^k
n-1 = 2k

Figure 5: (e)

5

(a)
Since the code is:
if A.length == 1 then
A[0] += 1;
else
int m = [A.length/2]; // $\frac{n}{2}$
Wacky(A.sub(0,m)); //this makes $T(\frac{n}{2})$
Wacky(A.sub(m,m)); //this makes $T(\frac{n}{2})$
Wacky(A.sub(0,1)); //this is constant 2

Then we can conclude recurrence relation is: $2T(\frac{n}{2}) + 2$
(b) Then from that relation a = 2, b = 2, f(n) = 2 g(n) $= n^{log_b a} = n^{log_2 2} = n$
Since $g(n) > f(n)$ which means g(n) become runtime O(n)