(1) $H_n = \sum_{k=1}^{n} \frac{1}{k}$ ← these terms are monotone decreasing as is the function $f(x) = \frac{1}{x}$ from 1 to n

so lower bound: $\int_1^{n+1} \frac{1}{x}\,dx = \ln(n+1) - \ln(1) = \ln(n+1)$

upper bound: usually we would do this from 0 to n but that wouldn't make sense here for $1/x$

So let's do $\sum_{k=2}^{n} \frac{1}{k} \le \int_1^n \frac{1}{x}\,dx = \ln(n) - \ln(1)$
$= \ln(n)$

and then add 1 to both sides so

$$\ln(n+1) \le \sum_{k=1}^{n} 1/k \le \ln(n) + 1$$

Since $\ln(n) + 1 - \ln(n+1) = 1 + \ln\left(\frac{n}{n+1}\right)$

$= 1 + \underbrace{\ln\left(1 - \frac{1}{n+1}\right)}_{\text{this term is negative but heads to zero as } n \to \infty} < 1$

So $\ln(n) \le H_n \le \ln(n) + 1$

so $H_n = \ln(n) +$ some small constant
$= \ln(n) + O(1)$ □

(2) and (3) are found in HW#3 (Probs 1 and 2) and HW#1 (Prob 6)

(4) $f_n = f_{n-1} + f_{n-2} \qquad f_0 = 0 \qquad f_1 = 1$

a) $a^n = a^{n-1} + a^{n-2} \qquad (\text{set } f_n = a^n)$

$$a^n = a^{n-2}(a+1)$$

$$a^2 = a + 1$$

$$a^2 - a - 1 = 0$$

$$a = \frac{1 \pm \sqrt{1+4}}{2} = \frac{1 \pm \sqrt{5}}{2}$$

So assume $f(n) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$

then $f(0) = 0 \Rightarrow c_1 + c_2 = 0$

$f(1) = 1 \Rightarrow \left(\frac{1+\sqrt{5}}{2}\right)c_1 + \left(\frac{1-\sqrt{5}}{2}\right)c_2 = 1$

So $c_2\left(\frac{1-\sqrt{5}}{2} - \frac{1+\sqrt{5}}{2}\right) = 1 \qquad (\text{substituting } c_1 = -c_2)$

$$-c_2\sqrt{5} = 1$$

$$c_2 = -1/\sqrt{5}$$

$$\Rightarrow c_1 = -c_2 = 1/\sqrt{5}$$

So $\boxed{f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n}$

b) $f_n = O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$

want to show: $\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n \leq c\left(\frac{1+\sqrt{5}}{2}\right)^n$

for some $c > 0$
$\forall\ n \geq n_0$

So $-\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n \leq \left(c - \frac{1}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n$

since $\left|\frac{1-\sqrt{5}}{2}\right| < \left|\frac{1+\sqrt{5}}{2}\right|$

then $\left(\frac{1-\sqrt{5}}{2}\right)^n < \left(\frac{1+\sqrt{5}}{2}\right)^n$

So we only need $\left|\frac{1}{\sqrt{5}}\right| \leq \left|c - \frac{1}{\sqrt{5}}\right| \rightarrow$ any $c \geq \frac{2}{\sqrt{5}}$.

$f_n = \Omega\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$

by induction: $f_n = f_{n-1} + f_{n-2} \geq c\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} + c\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}$

want $c\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} + c\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} \geq c\left(\frac{1+\sqrt{5}}{2}\right)^n$

$c\left(\left(\frac{1+\sqrt{5}}{2}\right) + 1\right) \geq c\left(\frac{1+\sqrt{5}}{2}\right)^2$

$\frac{3+\sqrt{5}}{2} \geq \frac{6+2\sqrt{5}}{4}$

$12 + 4\sqrt{5} \geq 12 + 4\sqrt{5}$ ✓

(5) $x < n \iff \lfloor x \rfloor < n$

a) Let $x < n$, then $\lfloor x \rfloor \leq x$ and $x < n$ so $\lfloor x \rfloor < n$.

b) Let $\lfloor x \rfloor < n$. If $x \geq n$, then since $\lfloor x \rfloor :=$ the largest integer $\leq x$ it follows that $n \leq \lfloor x \rfloor$. This would be a contradiction. So $x < n$. □

(6) a) [i] worst case of insertion-sort takes

$$f(n) = 1 + 2 + \cdots + (n-1) = \sum_{i=1}^{n-1} i = \frac{1}{2} n(n-1) \text{ comparisons}$$

So $T(1,000,000) = \frac{4 \text{ sec}}{\text{step}} \left(\frac{1}{2} (1\,000\,000)(999\,999) \text{ steps} \right)$

$= (500,000)(999,999)$ 4 secs $\left(\frac{1 \text{ sec}}{10^6 \text{ 4 sec}} \right)$

$= \frac{500,000}{1,000,000} \cdot (999,999)$ seconds

$\approx$ 500,000 seconds

$\approx$ **139 hours**

[ii]

best case for quicksort halves the array each time and checks $n-1$ times at each step so

$$T(n) = 2T(n/2) + (n-1) = (n-1) + 2\left(\tfrac{n}{2} - 1\right) + 4\left(\tfrac{n}{4} - 1\right) + \cdots + 2^{\lg n - 1}\left(\tfrac{n}{2^{\lg n - 1}} - 1\right)$$

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

So $T(1{,}000{,}000) = \left(1 \frac{\mu sec}{step}\right)\left(10^6 \lg(10^6) - 10^6 + 1\right)$ steps

$\leq 6 \cdot 10^6 \cdot \$ - 10^6 + 1 \quad \mu sec$

$= 23 \cdot 10^6 + 1 \quad \mu sec$

$= 23 + 1/10^6 \quad secs$

$\approx$ **23 seconds**

iii)

best case of ~~Kan~~ insertion-sort takes $n-1$ comparisons so

$T(10^6) = \left(1 \frac{\mu sec}{step}\right)(999{,}999)$ steps $\approx$ **1 second**

b) $$\frac{\left| 10^5 T_i(20) - 10^5 T_{ii}(20) \right|}{\frac{10^5 T_i(20) + 10^5 T_{ii}(20)}{200{,}000}}$$

$$= \frac{\left| 10^5 (n - n\lg n) \right| \cdot 200{,}000}{10^5 (n + n\lg n)} = \frac{200{,}000\,(20\lg 20 - 20)}{20 + 20\lg 20}$$

$$= \frac{200{,}000\,(\lg 20 - 1)}{\lg 20 + 1} \approx 3/5\,(200{,}000) = 120{,}000\ \mu sec$$

$= \underline{0.12 \text{ seconds}}$

(7)

integral upper bound:

$$\sum_{k=2}^{n-1} k \cdot \lg k \leq \int_2^n x \lg x \, dx$$

evaluate $\int x \lg x \, dx$ using integration by parts:

$u = \lg x$ $\qquad v = \frac{1}{2}x^2$

$du = (\ln 2)^{-1} \cdot \frac{1}{x} dx$ $\qquad dv = x\,dx$

$y = \lg x$

$2^y = x$

$y(\ln 2) = \ln x$

$(\ln 2)\, y' = \frac{1}{x}$

$y' = \frac{1}{x \ln 2}$

$$\int x \lg x \, dx = \frac{1}{2}x^2 \lg x - \int \frac{1}{2 \ln 2} x \, dx$$

$$= \frac{1}{2}x^2 \lg x - \frac{1}{4 \ln 2} x^2$$

So $$\sum_{k=2}^{n-1} k \lg k \leq \frac{1}{2}x^2 \lg x - \frac{1}{4 \ln 2} x^2 \Big|_2^n$$

$$= \frac{1}{2}n^2 \lg n - \frac{1}{4 \ln 2} n^2 - \frac{1}{2}(4) \lg 2 + \frac{1}{4 \ln 2} 4$$

$$= \frac{n^2 \lg n}{2} - \frac{n^2}{\ln(16)} - 2 \lg 2 + \frac{1}{\ln 2}$$

$$< \frac{n^2 \lg n}{2} - \frac{n^2}{\ln 16}$$

since $-2 + \frac{1}{\ln 2} < 0$

(8) a) $f(n) = n^\varepsilon$, $\varepsilon > 0$ $\quad g(n) = (\lg n)^4$

Let $c > 0$

then $(\lg n)^4 \le c n^\varepsilon$ for all $n \ge n_0$ (for some n_0)

since

$$\Updownarrow$$

$$\lg n \le \sqrt[4]{c} \cdot n^{\varepsilon/4}$$

since

$$\lim_{n\to\infty} \frac{\lg n}{\sqrt[4]{c} \cdot n^{\varepsilon/4}} = \lim_{n\to\infty} \frac{1/n}{\sqrt[4]{c} \cdot \frac{\varepsilon}{4} n^{\varepsilon/4 - 1}}$$

$$= \frac{1}{\sqrt[4]{c} \cdot \varepsilon/4} \lim_{n\to\infty} \frac{1}{n^{1+\varepsilon/4-1}} = \frac{1}{\sqrt[4]{c} \cdot \varepsilon/4} \lim_{n\to\infty} \frac{1}{n^{\varepsilon/4}} = 0$$

since $\varepsilon/4 > 0$.

So $g(n) = o(f(n))$.

b) yes, $g(n) = o(f(n))$ so $g(n) \neq \Theta(f(n))$.

c) yes. Let δ be any number in between 0 and ε, then as before $g(n) = o(n^\delta)$ and we know $n^\delta = o(n^\varepsilon)$.

So $n^\delta = o(n^\varepsilon - g(n))$.

(9) Let h be the height of each node of the heap. Then h ranges from 0 to $\lg n$ and the time max heapify spends on each node is $O(h)$. Level $\lg n$ has at most $n/2$ nodes and this decreases by a factor of 2 for each level we go up so

$$T(n) = \sum_{h=0}^{\lg n} \frac{n}{2^{h+1}} O(h) = O\left(n \sum_{h=0}^{\lg n} \frac{h}{2^{h+1}}\right)$$

$$= O\left(\frac{1}{2} n \sum_{h=0}^{\lg n} \frac{h}{2^h}\right) = O\left(n \sum_{h=0}^{\lg n} \frac{h}{2^h}\right)$$

$$= O\left(n \cdot \left[\frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \cdots + \frac{\lg n}{n}\right]\right)$$

$$= O\left(n \cdot \left(x + 2x^2 + 3x^3 + \cdots + \lg n\, x^{\lg n}\right)\Big|_{x=1/2}\right)$$

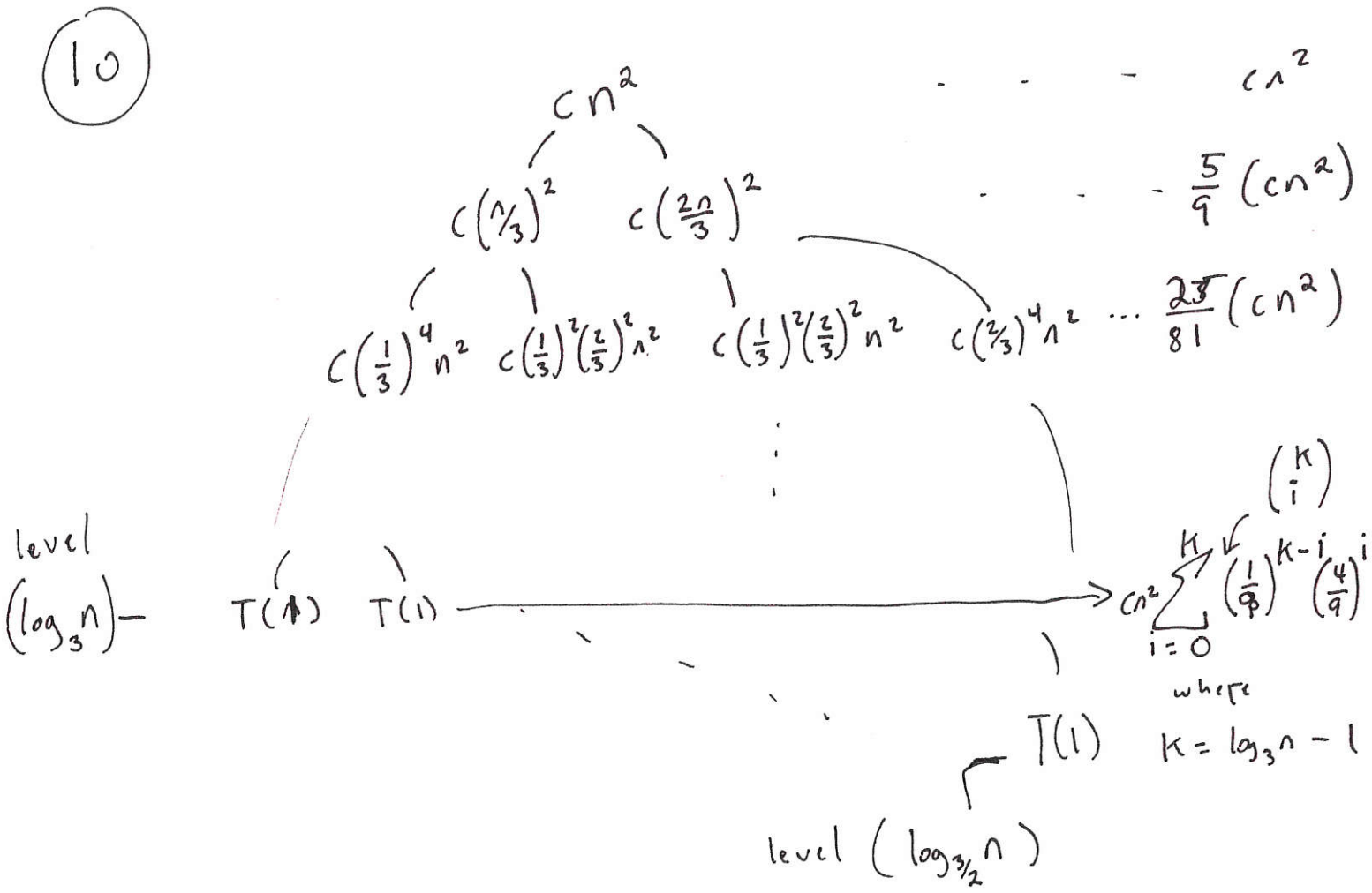
$$= O\left(n \cdot x \cdot \left(1 + 2x + 3x^2 + \cdots + (\lg n) x^{\lg n - 1}\right)\Big|_{x=1/2}\right)$$

$$= O\left(\frac{n}{2} \cdot \frac{d}{dx}\left(x + x^2 + \cdots x^{\lg n}\right)\Big|_{x=1/2}\right) = O\left(n \cdot \frac{d}{dx}\left(\frac{x^{\lg n + 1} - 1}{x - 1} - 1\right)\Big|_{x=1/2}\right)$$

$$= O\left(n \cdot \left[\frac{(\lg n + 1) x^{\lg n}(x-1) - (x^{\lg n + 1} - 1)}{(x-1)^2}\right]_{x=1/2}\right) = O\left(4n \cdot \left[(\lg n + 1) \cdot \frac{1}{n} \cdot -\frac{1}{2} - \frac{1}{2} \cdot \frac{1}{n} + 1\right]\right)$$

$$= O(-2\lg n - 2 - 2 + 4n) = O(n) \quad \square$$

10



each level (kth) for $k = 0, \ldots, \log_3 n - 1$

adds up to $cn^2 \sum_{i=0}^{k} \binom{k}{i} (\frac{1}{9})^{k-i} (\frac{4}{9})^i = cn^2 (1/9 + 4/9)^k$

this is binomial theorem

$$= (5/9)^k cn^2$$

So adds up to less than $\sum_{k=0}^{\log_{3/2} n - 1} (5/9)^k cn^2 + \left[\begin{matrix}\#\text{ of} \\ T(1)\text{'s}\end{matrix}\right] T(1)$

and more than $\sum_{k=0}^{\log_3 n - 1} (5/9)^k cn^2$

So we can use geometric series to bound above and below by

$$\left(\frac{(5/4)^{\log_3 n} - 1}{5/4 - 1}\right) cn^2 \leq T(n) \leq \left(\frac{(5/4)^{\log_{3/2} n} - 1}{5/4 - 1}\right) cn^2 + \left[\begin{matrix}\#\text{ of} \\ T(1)\text{s}\end{matrix}\right] T(1)$$

$$\frac{9}{4}\left(1 - n^{\log_3 5/4}\right) cn^2 \leq T(n) \leq \frac{9}{4}\left(1 - n^{\log_{3/2} 5/4}\right) cn^2 + \cdots$$

this is negative so this side is

$$[\text{constant}] \cdot n^2 - [\text{constant}] \cdot n^{2-\text{something}} = \Omega(n^2)$$

$5/4 < 3/2$ since $10 < 27$

so $\log_{3/2} 5/4 < 0$

so this side is

$$[\text{constant}] \cdot n^2 - [\text{constant}] \cdot n^{2-\text{something}} + [\text{constant}] \cdot \left[\begin{matrix}\#\text{ of} \\ T(1)\text{s}\end{matrix}\right] = O(n^2) + [\text{constant}] \cdot \left[\begin{matrix}\#\text{ of} \\ T(1)\text{s}\end{matrix}\right]$$

since $\left[\begin{matrix}\#\text{ of} \\ T(1)\text{s}\end{matrix}\right] \leq 2^{\left[\begin{matrix}\#\text{ of} \\ \text{levels}\end{matrix}\right]} \leq 2^{\log_{3/2} n} = n^{\log_{3/2} 2} < n^2$

since $\log_{3/2} 2 < 2$

$2 < (3/2)^2 = 9/4$

so $\Omega(n^2) \leq T(n) \leq O(n^2) \Rightarrow T(n) = \Theta(n^2)$ □

(11) Max-heapify compares a node to both of its children (if it has any), swaps it with the largest one if it is smaller and the continues comparing it down the tree or stops if it is bigger.

So the worst case is that a node's entry is smaller than everything below it and the algorithm moves it all the way down the tree.

So $T(n) = T(2n/3) + O(1)$

↖ = 2 if its the # of comparisons

by Master Thm this gives $T(n) = \Theta(\lg n)$

since $n^{\log_{3/2} 1} = \Theta(1)$.

(12) A sorting algorithm is stable if it leaves elements of the array w/ the same values in their original order.

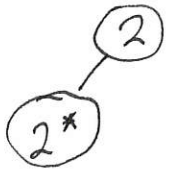
So, for example, $1, 2, 5, 1_{*}, 7, 1_{**}$

becomes $1, 1_{*}, 1_{**}, 2, 5, 7$

instead of $1_{*}, 1, 1_{**}, 2, 5, 7$

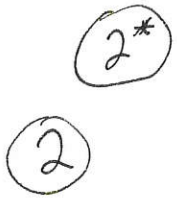
(* and ** mark the original 2nd and 3rd 1)

a) If I give HeapSort $\langle 2, 2^* \rangle$

then it runs build heap on 

which calls maxheapify on 2 which doesn't swap since the comparison looks for things strictly better than 2.

then it swaps and removes 2, then removes 2^*

$\Rightarrow$ 

and outputs $\langle 2^*, 2 \rangle$

So HeapSort (as written) is not stable.

b) quicksort is stable.

Let some number a be in positions i and j w/ $i < j$ of an array A and call them a and a^*
↑ starts in i ↑ starts in j

<u>case 1</u>: if $j < A.length$ then the run of quicksort will compare both a and a^* to the last element.

It compares from left to right in the array and swaps the element to a new position in the array if it is $\leq$ the last element otherwise it leaves them.

The positions that the elements swap into increases in index w/ each swap. So if $a \leq$ the last element then

they stay in the same relative positions - a listed before a^*. If they are bigger than the final element, then they stay put and the final element swaps (at the end) to a position ahead of both of them.

<u>case 2</u> : $j = A.length$

In this case a is $\leq a^*$ and will swap to some new position $k \leq i$ but at the end a^* swaps to at least $k+1$ so they remain in order.

Therefore, they remain in order for each call to Quicksort.

So quicksort is stable.

c) insert sort is stable since starting at 2 and going to the end, no element can move left past something of equal value since the comparison is strict.

(13) a) the first run of partition compares everything to the last element, changes nothing, and returns $q = A.length$ so Quicksort gets called again on an array just one shorter.

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

$= n + (n-1) + (n-2) + \cdots + 1$

$= \frac{1}{2} n(n+1)$.

b) I'm not exactly sure what this is asking, but if the question is whether a random array makes quicksort run faster than the worst case array, then sure because it could only get faster and at worst it would do the same.

c) I would add a running total of the differences and if it's zero at ~~the check~~ the end, then return $n/2$.

d) Partition (A, p, r)

$x = A[r]$

$s = 0$

$i = p - 1$

```
for j = p to r-1
    if A[j] ≤ A[r]
        s = s + (A[r] - A[j])
        i = i+1
        exchange A[i] with A[j]
    else s = s + (A[j] - A[r])
if s = 0
    return i+1
else return ⌊(r-p)/2⌋.
```

(14) was problem 7-3 from homework #3

(15) if Select divides the array into groups of 3, picks the median of each and then finds the median of medians, then x is bigger that (x) or equal to all medians below it and also the elements below them in their groups of 3.

So there are at least $\lceil n/3 \rceil - 2$ medians excluding x itself (the other 1 gets subtracted in case $n \not\equiv 0 \bmod 3$). And ceiling of half of these contribute 2 each that are $\geq x$.

So $2\left(\left\lceil \frac{1}{2}\left(\lceil n/3 \rceil - 2\right)\right\rceil\right) \geq n/3 - 2$ elements are

above x in the array.

And possibly up to $(n-1) - (\frac{n}{3} - 2) = \frac{2n}{3} + 1$ are below x. So this will be the worst case if we want ith value in there.

$$\text{So } T(n) \leq T(\lceil n/3 \rceil) + T(2n/3 + 1) + O(n)$$

Similarly for groups of 7 we get $4\left(\lceil \frac{1}{2}\left(\lceil n/7 \rceil - 2\right)\rceil\right) \geq \frac{2n}{7} - 4$

so

$$T(n) \leq T(n/7) + T\left(\frac{5n}{7} + 3\right) + O(n).$$

Solving: $n/7 \rightarrow T(n) \leq c\lceil n/7 \rceil + c(5n/7 + 3) + dn$

$$\leq \frac{6}{7}cn + 3c + dn = cn - \left(\frac{1}{7}cn - 3c - dn\right)$$

$$\leq cn$$

when $(\frac{1}{7}c - d)n - 3c \geq 0$

$n/3 \rightarrow T(n) \leq c\lceil n/3 \rceil + c(2n/3 + 1) + dn$

$$\leq cn + c + dn \overset{\leq}{\text{(circled)}} cn$$

Not True.
So this does not show $O(n)$