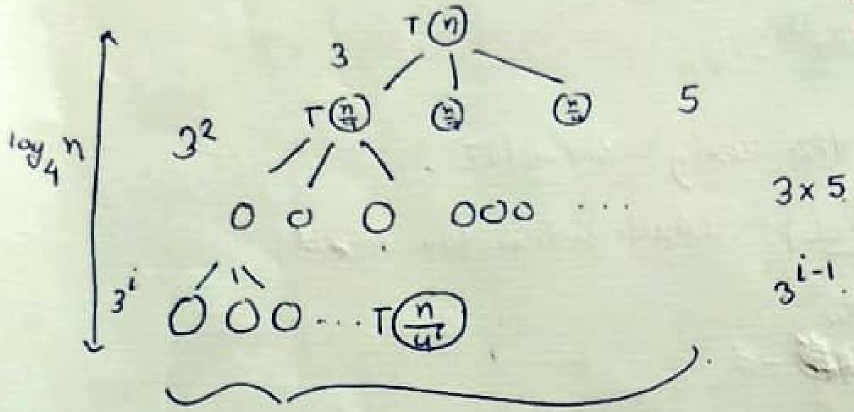
Anilha

MCS202304

(1)

~~87/100~~ 88/100

(6) (a) $T(n) = 3T(\frac{n}{4}) + 5$



now. $T(n) = 3T(\frac{n}{4}) + 5$

$$= 3^2 T(\frac{n}{16}) + 3 \cdot 5 + 5$$

$$\vdots$$

$$= 3^i T(\frac{n}{4^i}) + 3^{i-1} \cdot 5 + \dots + 3^1 \cdot 5 + 5$$

putting $4^i = n$ or, $i = \log_4 n$ we get,

$$= 3^{\log_4 n} \cdot T(1) + 5 \sum_{i=0}^{(\log_4 n)-1} 3 \qquad \text{(assuming } T(1) = 1$$

$$= 3^{\log_4 n} + 5 \cdot \frac{3^{\log_4 n} - 1}{3 - 1}$$

$$= 3^{\log_4 n} + \frac{5}{2} (3^{\log_4 n} - 1) \qquad [\, n^{\log_b c} = c^{\log_b n} \,]$$

$$= n^{\log_4 3}(1 + \frac{5}{2}) - \frac{5}{2}$$

So, a good estimate will be.

~~$T(n) = \Theta(n \log \dots)$~~ $T(n) = \Theta(n^{\log_4 3})$

$T(n) = \Theta(n^{\log_4 3})$

(2)

So, $T(n) \leq c(n^{\log_4 3}) - d$ for choice of $\left\{ \begin{matrix} d \geq 3 \\ c > d \end{matrix} \right\}$

$$T(n) = O(n^{\log_4 3})$$

18/18

TP $T(n) = \Omega(n \log_4 3)$

we have to show that

$T(n) \geq c' n \log_4 3$ for some constant c $\forall n \geq n_0$

Induction hypothesis :- $(T(n/4) \geq c' \frac{n}{4} \log_4 3$ $\Rightarrow T(n) \leq \frac{cn}{4} \log_4 3$

How do you get this?

$T(n) = 3T(n/4) + 5$

$$T(n) \geq 3c' \frac{n}{4} \log_4 3 + 5$$

$$= c' \left(\frac{n}{4}\right)^{\log_4 3} 4^{\log_4 3} + 5$$

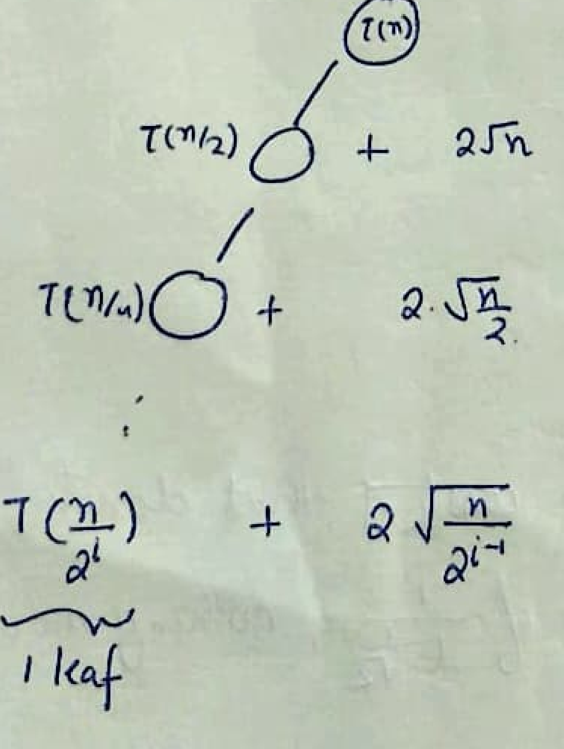
$$= c' n^{\log_4 3} + 5$$

$$\geq c' n \log_4 3$$ which is always true for $c \geq 1$, $n_0 \geq 1$.

$\therefore T(n) = \Omega(n \log_4 3)$

$\therefore T(n) = \Theta(n \log_4 3)$

(b) $T(n) = T(n/2) + 2\sqrt{n}$



So, $T(n) = \Theta(\sqrt{n})$ is a resonable estimate

RTP

now, $T(n) = O(\sqrt{n})$ and $T(n) = \Omega(\sqrt{n})$

∴ TP, $T(n) = O(\sqrt{n})$

we need to show that there exists constant c

such that.

Claim.

$$T(n) \leq c\sqrt{n} \quad \text{for all } n \geq n_0$$

Base case.

$$T(1) \leq c \quad \text{which is trivially true.}$$

Induction hypothesis

if $T(n/2) \leq c\sqrt{\frac{n}{2}}$

then $T(n) \leq c\sqrt{n}$

Proof:

$$T(n/2) \leq c\sqrt{\frac{n}{2}}.$$

$$\therefore T(n) = T(n/2) + \sqrt{n}$$

$$\leq c\sqrt{\frac{n}{2}} + \sqrt{n}$$

$$= c\sqrt{n} - (c-1)\sqrt{n} + c\sqrt{\frac{n}{2}}$$

$$= c\sqrt{n} - \sqrt{n}\left(c-1-\frac{c}{\sqrt{2}}\right)$$

$$= c\sqrt{n} - \sqrt{n}\left(c-\frac{c}{\sqrt{2}}-1\right)$$

now, To prove. $T(n) = O(n^{\log_4 3})$

we need to prove that. there exists a constant c and a constan[t]

such that; $T(n) \le c n^{\log_4 3} - d$ for all $n \geq n_0$

Claim: we have to prove that

now, $T(n) \le c n^{\log_4 3} - d$

We will prove this using induction.

Base case: $\boxed{T(1) \le c - d}$ which is true for $c \geq d$

Induction Hypothesis:

if $T(n/4) \le c\left(\frac{n}{4}\right)^{\log_4 3} - d$

then, $T(n) \le c n^{\log_4 3} - d$

Proof:

now.

$$T\left(\frac{n}{4}\right) \le c\left(\frac{n}{4}\right)^{\log_4 3} - d$$

$$T(n) = 3T\left(\frac{n}{4}\right) + 5$$

$$\le 3\left(c\frac{n}{4}^{\log_4 3} - d\right) + 5$$

$$= c \cdot \left(\frac{n}{4}^{\log_4 3}\right) \cdot 3 - 3d + 5$$

$$= c \cdot \left(\frac{n}{4}\right)^{\log_4 3} \cdot 4^{\log_4 3} - 3d + 5$$

$$= c n^{\log_4 3} - d - 2d + 5$$

$$= c n^{\log_4 3} - d - (2d - 5)$$

Now. if we choose $d \geq 3$.

$\hookrightarrow (2d - 5) \geq 0$

$\therefore T(n) \le c n^{\log_4 3} - d$ [for choice of $d \geq 3$]

$T(n) \leq c\sqrt{n}$ iff there exists a value of c such th

$$\left(c - \frac{c}{\sqrt{2}} - 1\right) \geq 0$$

or $c\left(1 - \frac{1}{\sqrt{2}}\right) - 1 \geq 0$

or $c \cdot P \geq 1$

or $c \geq 1/P$

or $c \geq \left(1 - \frac{1}{\sqrt{2}}\right)$

So, $T(n) \leq c\sqrt{n}$ holds for $c \geq \left(1 - \frac{1}{\sqrt{2}}\right)$

$\therefore T(n) = O(\sqrt{n})$.

---

now TP, $T(n) = \Omega(\sqrt{n})$

we have to show that, there exists a constant c st

<u>Claim</u>

$T(n) \geq$ ~~[illegible]~~ for all $n \geq n_0$

$c'\sqrt{n}$

Base case. ~~[illegible]~~ $T(1) \geq c$ ~~[illegible]~~ this is trivially true as we have freedom to choose c

Induction hypothesis

now if $T(n/2) \geq c'\sqrt{n/2}$ ~~[illegible]~~ let c' be 1

then, $T(n) \geq$ ~~c~~$\sqrt{n}$

$T(n) = T(n/2) + 2\sqrt{n}$

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

$> \sqrt{n}$

$\therefore T(n) = \Omega(\sqrt{n})$

$T(n) = \Theta(\sqrt{n})$

2.

5/15

now, $T(n) = T(n/2) + 2\sqrt{n}$

$$= (T(\tfrac{n}{4}) + 2\sqrt{\tfrac{n}{2}}) + 2\sqrt{n}$$

$$\vdots$$

$$= T(\tfrac{n}{2^i}) + 2\sqrt{n}(1 + \cdots \tfrac{1}{(\sqrt{2})^{i-1}})$$

$$\vdots$$

if $2^i = n$ or $i = \log_2 n$ then $= T(1) + 2\sqrt{n}\,(1 + \cdots \frac{1}{(\sqrt{2})^{\log_2 n - 1}})$

$$= 1 + 2\sqrt{n} \sum_{i=0}^{\log_2 n - 1} \frac{1}{(\sqrt{2})^i} \quad [T(1) = 1]$$

now, $\sum_{i=0}^{\log_2 n - 1} \frac{1}{(\sqrt{2})^i}$

$$= \frac{1 - (\frac{1}{\sqrt{2}})^{\log_2 n}}{1 - \frac{1}{\sqrt{2}}}$$

$$= \frac{1 - (\frac{1}{2})^{\log_2 \sqrt{n}}}{1 - \frac{1}{\sqrt{2}}}$$

$$= \frac{1 - (\sqrt{n})^{\log_2 \frac{1}{2}}}{1 - \frac{1}{\sqrt{2}}}$$

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

$$= p \cdot (1 - \tfrac{1}{\sqrt{n}})$$

[Here p is constant that den... value of $\frac{1}{1 - \frac{1}{\sqrt{2}}}$, nothing ... any time, p and $\frac{1}{1 - \frac{1}{\sqrt{2}}}$ are e...

$\therefore T(n) = 1 + 2\sqrt{n}\,(p)(1 - \frac{1}{\sqrt{n}})$

$$= (1 + 2p\sqrt{n} - 2p)$$

Input: $n \geq 1$, size of the string

S, the string

output: number of different pairs (i, j), $0 \leq i < j \leq (n-1)$ such that

$(s[i] \ldots \ldots s[j])$ is palindrome

(a)

Algorithm:

```
Count Pal Substrings (S, 0, n-1):
    count = 0
    for (i=0; i ≤ n-1, i++):
        for (j= i+1; j ≤ n-1, j++):
            if (is Palindrome (S, i, j) is True):
                count ++;
    return count,
```

~~isPalindrome (S, i, j)~~

```
is Palindrome (S, l, r):
    if (l ≥ r) return True,        // Base case
    if S[l] ≠ S[r] return false;
    return isPalindrome (S, l+1, r-1)
```

$\frac{18}{18}$

now we are iterating over every element / character of S

and for that char we are iterating over, all the posterior

characters. So, we are going through all possible substring

S[i].. S[j] where $i = 0(1)(n-1)$

$j = (i+1)(1)(n-1)$

is set of all non-empty substrings of S.

we are taking them and increasing the count if it's a palin

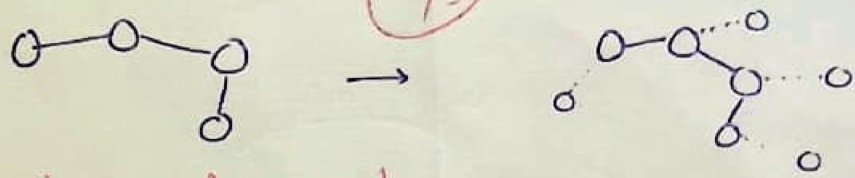
drome.

So, our algorithm should be correct.

15/15

1) Let's have this example:

(4.b) (G) 0/5 (G') = { }

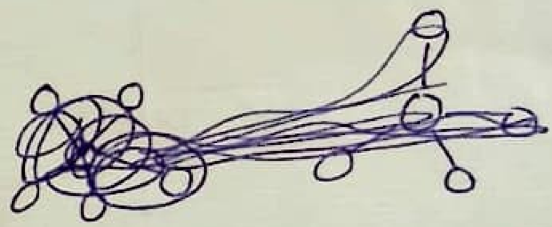


This is not a counter-example.

from G we are getting G' which seems to be connect, but the problem is, it only consist of 4 graphs directly derived from G, but G(5,4) also consists of many more graphs that are acyclic, has no isolated vertices, 4 edges, 5 vertices

one such G'' be..

G'' =



0/5

~~So, we should prove that if G' exis~~

but, if we assume G to be , we can eas derive G' from the stated rule.

~~So, G'~~ to prove induction on G, we should take

Why?.

a vertex with G'(n+1, n), remove any of the vertex → G (n, n-1)

and show that if G holds the properties, adding an edge/vertex G → G' will also hold the property.

Base Case: what

if ~~[illegible]~~ (Heigh... != Height(y):

then going to the same level by traversing the path have t...

from $x/y \to r$ will ~~[illegible]~~ result in same level eventually

why?

after that, if $x==y$, then that's the LCA, Done!

~~let ...~~

Induction hypothesis:

~~if ... v is the LCA of (x,y)~~

~~then v is also the~~

0/10

if $v = LCA(par(x), par(y))$ and $x \neq y$,

then $v = LCA(x, y)$

This is not an indu... hypothesis! What exact...

Also not true: consider:

are you inducting o...

Proof. let v is the $LCA(par(x), par(y))$

then v is the first vertex that is common from both the path

$par(x) — .. v — r$, $par(y) ...... v \to r$ ~~par(x), par(y) → r~~

now if we go the children then from x ~~to~~, y to r path will ...

$x \to par(x) — .. v .. r$ ~~[illegible]~~ case $x \neq y$.

$y \to par(y) .. v .. r$ v is the first common ver...

(Proved)

(b) LCA (x, y):

→ while (Height (x) > Height (y)):

→ → $x =$ par (x)

→ while (Height (y) > Height (X)):

→ → $y =$ par (y)

→ if $(x == y)$ return x

→ else return LCA(Par(x), Par(y))

Height( ):

$\frac{15}{18}$

So, $T(n) = \Theta(\sqrt{n})$ is a resonable estimate

RTP

now, $T(n) = O(\sqrt{n})$ and $T(n) = \Omega(\sqrt{n})$

$\because$ TP, $T(n) = O(\sqrt{n})$

we need to show that there exists constant c

such that.

Claim $T(n) \leq c\sqrt{n}$ for all $n \geq n_0$

Base case. $T(1) \leq c$ which is trivially true.

Induction hypothesis

if $T(n/2) \leq c\sqrt{\frac{n}{2}}$

then $T(n) \leq c\sqrt{n}$

Proof: $T(n/2) \leq c\sqrt{\frac{n}{2}}$

$$\begin{aligned}
\therefore T(n) &= T(n/2) + \sqrt{n} \\
&\leq c\sqrt{\frac{n}{2}} + \sqrt{n} \\
&= c\sqrt{n} - (c-1)\sqrt{n} + c\sqrt{\frac{n}{2}} \\
&= c\sqrt{n} - \sqrt{n}\left(c-1-\frac{c}{\sqrt{2}}\right) \\
&= c\sqrt{n} - \sqrt{n}\left(c-\frac{c}{\sqrt{2}}-1\right)
\end{aligned}$$

Height (x):

~~return~~ if x. parent = NIL

→ → return 0

else return [1+ Paren (x)]

How do you add an object (or its id) to 1 and have it make sense?

Parent (x):

return ~~A~~ arr obj [x. par]

stores all the vertices objects

such that [arr obj [i] = v if v. id = i]

(c) Proof of correctness:

first some intuition, ~~if~~ if x, y are not in same level [i.e Height (x) ≠ Height (y)] then we should make them at same level and that won't change the answer. Why?.

Contradiction.

(b) LCA (x, y):

→ while (Height (x) > Height (y)):

→ → x = par (x)

→ while (Height (y) > Height (X)):

→ → y = par (y)

→ if $(x == y)$ return x

→ else return LCA(Par(x), Par(y))

$\frac{18}{18}$

~~Height(x~~):

(b) We have to prove that, the above algorithm is correct.

lets first prove that.

is Palindrome (s, l, r) is correct

Base case: if $l \geq r$,

either string is empty or contains a single letter

thats a palindrome

Inductive hypothesis.

if is Palindrome $(s, l+1, r-1)$ returns

correctly, so does isPalindrom (S, l, r).

Proof:

if $S[l] = S[r]$

then the control reaches to Is Palindrome $(S, l+1, r-1)$

which returns correctly.

if it returns true, so does the prior step.

if $S[l] \neq S[r]$

then it's not a palindrome, returns false

Induction hypothesis:

if $[(q, r) = \text{QuotRem}(x-y, y)] \Rightarrow qy + r = (x-y)$

then $[(q+1, r) = \text{QuotRem}(x, y)] \Rightarrow (q+1)y + r = x.$

Proof

$$(q, r) = \text{QuotRem}(x, -y, r)$$

$$\therefore\ (qy + r) = x - y$$

now our algorithm on running input $\text{QuotRem}(x, y)$

gives $(q+1, r)$ where $(q, r) = \text{Quotrem}(x-y, y)$

now

$$\left.\begin{aligned} (qy + r) &= x - y \\ \text{or}\quad qy + y + r &= x \\ \text{or}\quad (q+1)y + r &= x \end{aligned}\right\}$$

$$\therefore\ \text{Quotrem}(x, y) = (q+1, r).$$

You have not stated what you are inducting on.

I don't see any induction happening here.!

0/5

(a)

Prove that $LCA^{(x,y)}$ exists:

Assumption : LCA (x, y) does not exist.

that implies, there is no ~~common~~ v in the T that is

common in the path of $x \to r$ and $y \to r$.

Contradiction :

But that's a direct contradiction as, r is common in the path of

both of them.

~~Lethy~~

So, LCA (x, y) exists

~~3/5~~

3/5

(b) Prove that $LCA^{(x,y)}$ is unique :

~~Assumption : let v_1, v_2 be the vertices such that~~
~~both of them are the LCA (x, y) and $v_1 \neq v_2$~~
~~let's consider two paths $v_1 \to r$, $v_2 \to r$~~
~~now $v_1 \to r$ is a subpath of~~

Assumption. let v_1, v_2 be the vertices, such that

$v_1, v_2 = LCA(x, y)$ and $v_1 \neq v_2$.

Both v_1, v_2 belongs to path $x \to r$..

QuotRem (x, y):

if $(x-y) < 0$: return $(0, x)$,

else

→ (q, r) = QuotRem $(x-y, y)$

→ return $(q+1, r)$

5/5

(b) Correctness Proof:

claim. QuotRem $(x, y) = (q, r)$

iff $x = qy + r$ [where $(0 \leq r < y)$]

What are you inducting on?

Base case: if $(x-y) < 0$.

or $x < y$.

$x = 0y + x$ [we can write this because $x < y$]

$= qy + r$

where $q = 0$, $r = x < y$.

∴ it return $(0, x)$ which is true

③ Page

(4) (a) Algorithm

Search (n, A, v):

for ($i=0, i \le n-1$; i++)!

→ if (A[i] == v)

→ → return i

return "NIL"

input: (n) number of elements in array A

(A) the input Array

(v) the search term

output: {if v appears atleast 1 then return that index} {else return "NIL"}

10/10

We are considering the universe, where addition, (sub, mult...), assignment, return, comparison takes constant amount of time

(c) Search (n, A, v):

for ($i=0, i \le n-1$, i++)	$\sum_{j=1}^{K} T_j$	[where, K is the index where v appears for the first time, otherwise n] if v does not appear
→ if (A[i] == v)	$\sum_{j=1}^{K} T_j$	
→ → return i	1	
return "NIL"	1	

What is T_j? What is j?

now, if we consider n as the recurrence parameter

$$T(n) = \sum_{j=1}^{K} 2T_j + 2$$

now in the worst case, v does not appear in the array, in that case $K = n$, as we have to loop through every element of A considering $t_j = 1$ What is j?

$T(n) = 2n + 2$

So, TP $T(n) = O(n)$

there exists c, n_0 such that

we need to prove that $T(n) \leq cn$ for all $n \geq n_0$

now ~~T~~

we choose constant c to be 3

and n_0 to be 2.

or $T(n) \leq 3n$ for all $n \geq 2$

now we need to prove this

Hence, $T(n) - 3n = (2n+2) - 3n$

$= 2 - n$

for $T(n) \leq 3n$ to hold for all $n \geq 2$

$T(n) - 3n \leq 0$ for all $n \geq 2$

$\therefore$ which is true

as $(2-n) \leq 0 \ \forall n \geq 2$

8/10