CS213 Sumit Jain
190050119

EndSems

Question -1

i)

Let s_1 be the string such that it has length n, and $f(s_1)$ is undefined,

Then, The only possible case is when
$S_1 = S_2 \circ 01$ for a string S_2 in S

Otherwise, if it is of the form $S_2 \circ 0$, then, whether $g(s_2)$ is defined or not, $f(s_1)$ would always be defined.

Now, in order for S_1 to be undefined,
$g(S_2)$ must be undefined

$\Rightarrow$ S_2 must end at 0 for it to be valid

$\Rightarrow S_2 \Rightarrow S_3 \circ 0$ S_3 is of length $n-3$
such that $f(s_3)$ is not defined

Now again, it has to have 01 at last for it to be undefined.
So, this gives us a form for s_1
such that

if $n \bmod 3 \Rightarrow 0$ then $S_1 = \underbrace{001}\underbrace{001}\ldots$ $\hookrightarrow n/3$ times

if $n \bmod 3 \Rightarrow 1$ then its of the form $1 \circ s'$
where $s' = \underbrace{001}\underbrace{001}\ldots$
001 repeated $\frac{n-1}{3}$ times

If $n \bmod 3 = 2$, $s_1 = 01 \cdot s'$

where $s' = \underbrace{001}\ldots$ repeated $\frac{n-2}{3}$ times

$\Rightarrow$ if $n \bmod 3 = 0$

1's occurs at positions multiple of 3

0 0 1 0 0 1 . . . .
(3) (6) . . .

if $n \bmod 3 = 1$

1's occur at positions ~~multiple of~~ $3K+1$, where K is $\{0, \ldots \frac{n-1}{3}\}$

1 0 0 1 0 0 1.
(1) (4) . . .

if $n \bmod 3 = 2$

1's occur at positions $3K+2$ where $K \in \{0, \ldots \frac{n-2}{3}$

0 1 0 0 1 0 0 1 . . .
(2) (5) . . .

Now, similarly for $g(s_1)$ is undefined only when it ends at 0.

$\Rightarrow$ $s_1 = s_2 \cdot 0$

$\Rightarrow$ Now $f(s_2)$ must be undefined,

s_2 is $n-1$ string, from previous analysis for $f(s)$ we have seen that exactly one such string $s_2 \in S$ of length $n-1$ s.t. $f(s_2)$ is undefined

$g(s_1)$ is undefined iff $f(s_2)$ is undefined

$\Rightarrow$ and $s_1 = s_2 \cdot 0$

ii) Let's apply induction on length of strings

~~Say let's say for~~ When length = 1

~~We~~ Base case :-> $\cancel{g} f(g(1)) = f(0) = 1$

$\Rightarrow$ condition holds that $\cancel{g} f(g(s)) = s$

for all $s \in S$ with length 'k' such that

$g(s)$ is defined.

similar $g(f(0)) = g(1) = 0$

So, let's say that $s_1 \in S$ are all strings with a particular length 'k' such that $g(s_1)$ is defined

then, $f(g(s_1)) = s_1$ and $s_2 \in S$ such that

~~$len(s_2) = k$, and $f(s_2)$ is defined~~

~~then $g(f(s_2)) = s_2$~~

We have proved that for length $k = 1$

(Base case) this holds.

Now let's say that this holds for all

$K \leq n$ (i.e. for all strings, s_1 and s_2

with $1 \leq$ length $\leq n$

(Inductive hypothesis)

~~Now let's say we have to prove that~~

Now, given base case, and inductive hypothesis, let's prove this for all s_1, s_2 s.t. $g(s_1)$ and $f(s_2)$ is defined and $len(s_1) = len(s_2) = n+1$

$len(s) = n+1$

Let s be a string $len(s) = n+1$ and $g(s)$ is defined.

s can be $\Rightarrow s'\cdot 0$ where $len(s') = n$

$s'\cdot 01$ $len(s') = n-1$

Case 1: $s = s'\cdot 0$ $len(s') = n$

Now, $f(g(s)) = f(g(s'\cdot 0))$

$g(s'\cdot 0) = f(s')\cdot 0$ if $f(s')$ is defined

else undefined (this case does not arise as $g(s)$ is defined)

$\Rightarrow f(s')$ is defined and $f(g(s',0))$

$= f(f(s')\cdot 0) = g(f(s'))\cdot 0$ if $g(f(s'))$ is defined

$\therefore$ By induction hypothesis if $f(s')$ is defined then, $g(f(s'))$ is defined and

$\cdot$ $g(f(s')) = s'$ ($\because len(s') = n$)

Cas

$f(f(s').0) = g(f(s')).0$

$= s'.0$ Hence Proved.

$f(g(s)) = s$

Case 2: $\to$ $s = s'.01$, $len(s') = n-1$

$g(s'.01) = f(s').01$ if $f(s')$ is defined,

$g(f(s')) = s'$ (By induction hypothesis)

$f(g(s'.01)) = f(f(s').01)$

$g(f(s')).01$

from induction hypothesis,

$\Rightarrow s'.01$

~~Now~~ suppose for

if $f(s')$ is not defined,

$g(s'.01) = s'.0.0$

$f(s'.0.0) = g(s'.0).0$ if $g(s'.0)$ is defined.

~~But if f(s') g(s'.0) = f(s').0 if f(s') is~~

But since, ~~g~~ $f(s')$ is not defined. $g(s'.0)$ is not defined

so, $f(s'.0.0) = s'.0.1 = s'.01$

$= s$

$f(g(s)) = s$

since, we covered both possibilities for

$s = s'.01$, i.e. either $f(s')$ is defined

or not,

Also, we proved for both case, whether

$s = s'.01$ or $s'.0$ there is

no other possibility

And it takes all cases of s, as s' is arbitrary of $n-1$ and n respetively in two cases.

similarly, let s be of $len(s) = n+1$,

and $f(s)$ is defined

then Case 1: $s = s'.0$

$\Rightarrow g(f(s)) = g(g(s').0)$

$f(s) = g(s').0$ if $g(s')$ is defined

$g(g(s').0) = f(g(s')).0$

$\Rightarrow s'.0$ (from induction hypothesis)

if $g(s')$ is not defined

$\Rightarrow f(s'.0) = s'.1$

$g(s'.1) = g(s''.01)$ s.t. $s' = s''.0$

and $f(s'')$ is not defined

then $g(s''.01) = s''.00.0$

$= s'.0$

$= s$

$g(f(s)) = s$ in this case as well

$g(s'')$ is not defined iff $f(s'')$ is not
(defined proved earlier.)

So, we have covered both cases for $s = s'.0$
where $g(s')$ was defined and also
where it was undefined.

Now, last case 4: $s = s'.01$ and
$f(s)$ is defined

$g(f(s'.01)) = g(g(s').01)$
$\because f(s)$ is defined $\Rightarrow g(s')$
will be defined

$g(g(s').01) = f(g(s')).01$
$\because g(s')$ is defined $\Rightarrow f(g(s')) = s'$
By inductive hypothesis

$\Rightarrow g(g(s').01) = s'.01$

$\Rightarrow g(f(s'.01)) = s'.01$

$\Rightarrow g(f(s)) = s$

Therefore by strong induction ~~B~~ we have proved that for all $s_1 \in S$ s.t. $len(s_1) = n+1$ and $f(s_1)$ is defined
$g(f(s_1)) = s_1$.

and ~~of~~ $s_2 \in S$, s.t. $len(s_2) = n+1$
and $g(s_2)$ is defined
$f(g(s_2)) = s_2$

iii) Base case, $s_2, f(s_2), f(f(s_2))$..
$len(s_2) = 1$
$\Rightarrow$ $g(0)$ is undefined
$s_2 = 0$
a, $f(0)$
$\Rightarrow 0, 1,$ $f(1)$ is undefined

also 0, 1 are all possible strings of ~~length~~ '1' in S

Now, assume, this holds for $k \leq n$ for an arbitrary n.

now, for $n+1$

say $len(s) = n+1$,
$g(s)$ is undefined
only possible when $s = s'.0$
for some s'

$\Rightarrow$ ~~g~~ $s = s'.0$

$f(s'.0) = g(s').0$ else $s'.1$
if $g(s')$ is defined

So this gives all the possible
~~g(s')~~ strings of length $n+1$

$f(~~f(~~g(s').0) = g(g(s')).0$
if it is defined
by inductive hypothesis

This gives all the possibilites, as for any position it considers both bit being 0 and 1,

For e.g. $f(s'.0) = g(s').0$ if $g(s')$ is defined

~~We know that~~ it would cover all cases
for strings with length $n+1$ ~~cap.~~ ending in 0.

Similarly $f(s'.0) = s'.1$ if $g(s')$ is undefined

$\Rightarrow$ s' can take all possible strings which have length n

and thus ~~s~~
s is string ending in 1 with length $n+1$.

~~These c~~

~~This covers both possibities.~~

Case 2:

Similarly, if ~~s is~~ len(s) = n+1

and f(s) is undefined

$\Rightarrow$ ~~f(s)~~ s ends in '01' proved in i)

$\Rightarrow g(s) = f(s').01$ if $f(s')$ is defined

else $s'.0.0$

it also covers both cases of string of length n+1 ending with either ~~...~~. 1 or 0.

Question 3

3) i). ~~n=~~ $n=1$, we have only one node, which is the required node.

$n=2$, we have two nodes, the root node is the required node

Now $n=3$, we have root node which satisfies the condition s.t. No. of descendants d, $\frac{n}{3} < d < \frac{2n+1}{3}$

Now, for $n \geq 3$,

~~For every node~~ Now starting at root node we can traverse down the tree.

if at an instance curr-root defines the pointer to current root, then it's either that the node is a solution ~~or it's children will be a solution~~ or one of its descendent will be a solution, if ~~it is~~ curr-root is a solution ~~we have too~~

$\Rightarrow$ we found a node with descendants lying b/w $\frac{n}{3}$ and $\frac{2n+1}{3}$

descendant = d

if curr-root is not a solution this means it has ~~nodes >~~ descendants $> \frac{2n+1}{3}$

$\Rightarrow$ one of its chidren definitely has atleast $n/3$ ~~nodes~~ descendants, because

because ~~one of de~~

descendants$_{curr-root}$

$=$ descendant$_{right}$ $+$

descendat$_{left}$ $+ 2$

if left, right both exists else, = descendant exists + 1

$\Rightarrow$ traverse down to child with descendant $(d) > n/3$, ~~Now if this child has~~

~~more tha~~ descendants $< \frac{2n+1}{3}$

we could continue to iterate till we get descendant $< \frac{2n+1}{3}$

since, no. of descendants always decrease atleast by 1 in this process, therefore, it will terminate

And curr-root will be the required node.

Take the case when

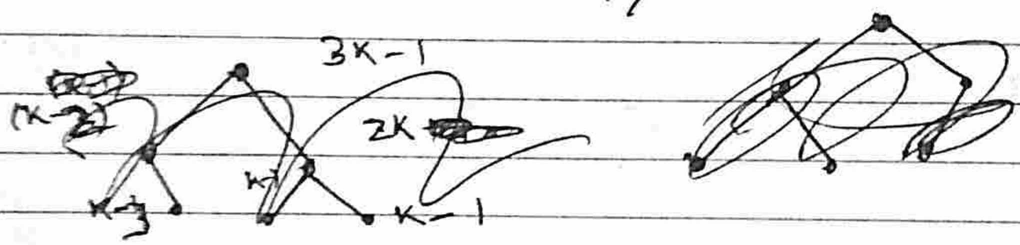
$n = 3K-1 \quad \forall K \in \{2, 3, \ldots\}$

descendants $< \frac{n+1}{3} = K$

or

descendants $> \frac{2n}{3} = \frac{6K-2}{3}$

descendent $\geq 2K$

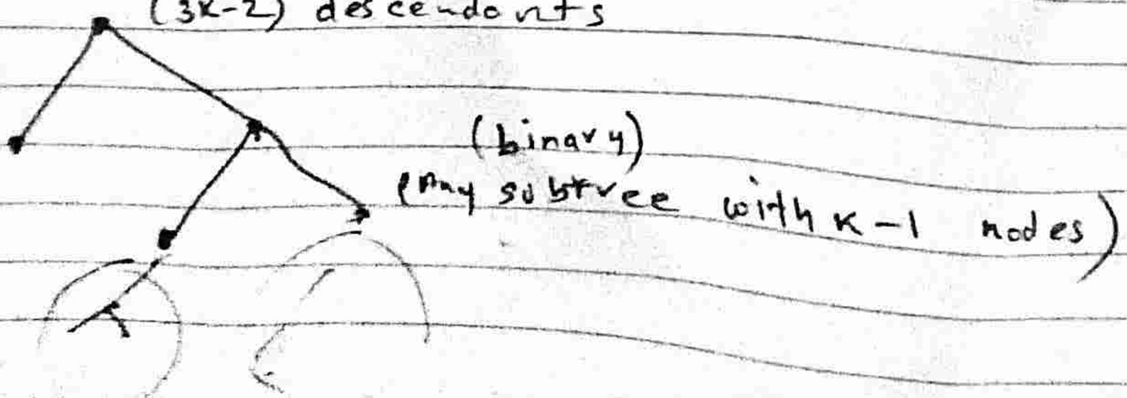


root node $\Rightarrow$ $(3K-2)$ descendants
left_chid $\Rightarrow$ $K-1$ descendants
Righ_child $\Rightarrow$ $2K$ descendants

Right_child have two children with $K-1$, $K-1$ descendant each.

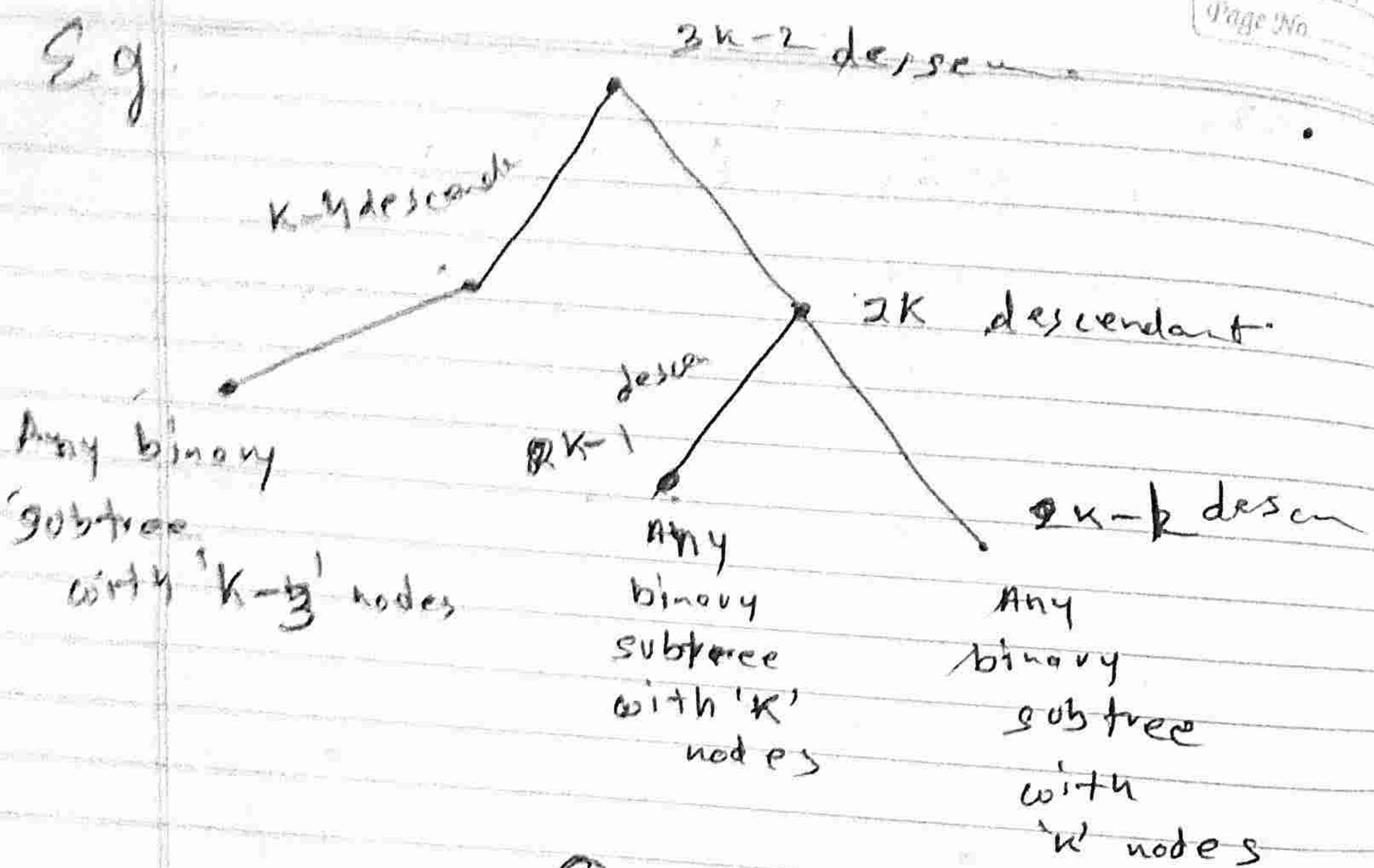
They could be any subtree since, now ~~root~~ subtree will not have descendents greater than 1

(3K-2) descendants



$n = 3k-1 \quad k \in N$

Eg



Question-4

ii) Smallest such undirected graph would be K_4 i.e. complete graph with 4 vertices
it has no two disjoint odd cycles.