Name - Varun Gupta
Roll No - 2023101108
Course - AAD

Problem Set - 1

A2)

To Proove :- For a graph G on n node (n is a even no.).
(or Disproove) If every node of G has a degree at least n/2, then G is connected.

Proof :- Let for the sake of contradiction G is not connected i.e it is disconnected. It means G can be broken down int two disjoint non empty vertex sets A & B such that any vertex ∈ A is not linked with any vertex ∈ B.

$|A| = a \quad (a \geq 1)$

$|B| = b \quad (b \geq 1)$

For A,

Each vertex deg $\geq n/2$

$$\text{Total deg}_A \geq a \cdot \frac{n}{2}$$

$$\text{Maximum } \deg(v)_A = (a-1)$$

$$\text{Max Total degree} = a \cdot (a-1)$$

Now;

$$a(a-1) \geq a \cdot \frac{n}{2}$$

$$\Rightarrow a \geq \frac{n}{2} + 1$$

Similarly,

$$b \geq \frac{n}{2} + 1$$

But

$$a + b = n$$

$$a + b \geq 2\left(\frac{n}{2} + 1\right)$$

$$n \geq n + 2$$

$\Rightarrow$ This is impossible which contradicts the assumption that G is ~~di~~ not connected.

Hence Proved.

Ans 3)

(a) To Proove :- In any Binary Tree the number of nodes with two children is exactly one less than no of Leaves.

Proof :- Let 'I' be total No of Internal Nodes and 'L' be Leaf Nodes.

Now $\boxed{I + L = N}$ — (1)

Now Let K be total no of Internal Nodes with exactly 2 child.

Total edges = N - 1

Edges ~~with~~ by node with 2 child = 2

" " " " 1 " = 1

" " leaf " = 0

Equating Total edges,

$\therefore N-1 = K(2) + 1 \times (I-K)$

$N-1 = I+K$

from (1),

$N-1 = N-L+K$

$\boxed{L-K=1}$

Hence Prooved.

(b) ~~To Proove:~~ For a Full Binary Tree,

Using above Notation (from part a);

$K = I$

$K + L = N$

also, $L - K = 1$

From these,

$(L-1) + L = N$

$\boxed{L = \frac{N+1}{2}}$ Ans

Ans 4)

(a) To Proove :- WFS marks every vertex reachable from s and only those.

Proof :-

For any vertex 'v' to be marked by WFS it need to get processed by WFS or come in bag.

Claim :- Any vertex which is reachable from s will be marked by WFS.

Proof :- Let take vertex 'v' reachable from s.

As it is reachable, there will be a path from s to v.

Let it be, $s\ v_1\ v_2 \ldots v_k\ v$.

Now in algo,

first s is taken from and we put its ~~adjacent~~ neighbour(u)

Similarly v_1 will be taken out and it put v_2.

Similarly in this way v will be added in bag and marked.

Claim :- No vertex which is ureachable from s will not be marked by WFS.

Proof :- Let a vertex 'u' not reachable from s i.e their does not exist a path from s to u.

As there is no path, this means no neighbour of u is neighbour to any neighbour of s.

As WFS, only process neighbour's of s.

∴ 'u' will never be processed.

Hence from both claims our assertion is prooved.

(b)

(i) To Proove :- No black vertex is a neighbor of a white Vertex.

Proof :- Lets prove using Induction.

Base - Initially all vertices are white so it holds.

Let assume the invariant holds till some step 'k' in algo. such that $1 \leq k$.

K+1th Step

- A gray vertex 'v' is choosen.
- If ~~it~~ v does not has any white neighbour then will color 'v' black. Invariant holds because vertex v which is black do not have any white neighb.

- Else :- we choose any white vertex say 'w' and color it gray. Since both v & w are grey there is no black - white relation.

* Also no previous has it because invariant holds upto K steps. Thus here also invariant holds.

Therefore it is true for $k+1^{th}$ step.

Since k is arbitary no. ∴ it is true for all steps.

Hence Prooved.

(ii)

1] For Reachable Vertices :-

* From Invariant, when a vertex is black it has no white neighbours. Since algo only colors neighbours of gray vertex gray. Therefore neighbors of black are not considered. When all neighbours of gray vertex is colored it turns black.

  ∴ all vertices reachable from s will eventually become black.

2] For Unreachable vertex :-

The algorithm only processes neighbour of a vertex. Therefore vertex not reachable from s will never be visited and hence remains white.

3] For Spanning tree :-

We know spanning tree is a subset of a connected undirected graph that includes all the vertices of the orignal graph without any cycles.

Here each vertex has exactly one parent except s which create a tree rooted at s [no cycle].

As algo runs, tree will grow to include all reachable vertex making it a spanning tree.

Hence Prooved.

Ans 1) Given:-

$T_1(n) = a \cdot T_1\left(\frac{n}{b}\right) + bn$

$T_2(n) = b\, T_2\left(\frac{n}{a}\right) + an$

$a \geq b$ and $T_1(1) = T_2(1) = 1$.

Assumption:- $a \geq 1$ & $b > 1$

Proof:- By using Masters theorem,

$$T(n) = a\, T(n/b) + \Theta(n^k \log^p n)$$

For T_1:-

$a = a,\ b = b,\ p = 0,\ k = 1$

As $a > b^k$ $\therefore$ $T(n) = \Theta(n^{\log_b a})$

For T_2:- $a = b,\ b = a,\ p = 0,\ k = 1$

there $T_2 = \Theta(n^k \log^p n)$

$= \Theta(n)$

Since $a \ge b$

$\frac{a}{b} \ge 1$

$\Rightarrow \log_b a \ge \log b$

$\log_b a \ge 1$

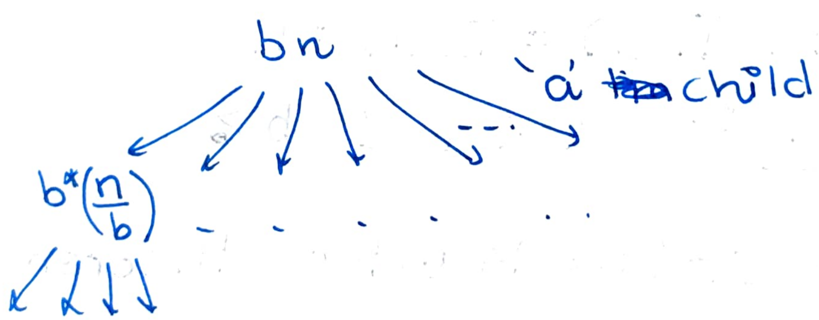
$\Rightarrow T_1 \ge T_2$ for large value of n.

Alternate:

If our assumption is not true then :-

we use recursion tree to analyse.

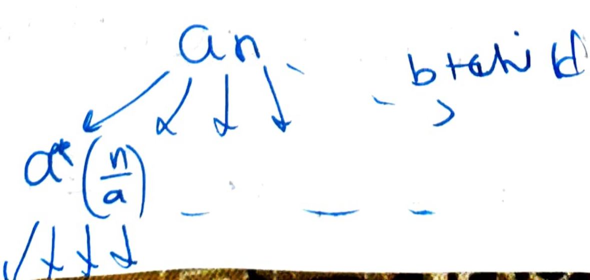
for T_1



The height would be $\log_b n$.

Total level cost $= a * b\left(\frac{n}{b}\right) = an$ (except root)

for T_2



Total height = $\log_a n$

" cost each level = $b * a\left(\frac{n}{a}\right) = bn$

Comparing T_1 & T_2

As these both function are ~~comp~~ computing cost we can compare them by comaring their total cost.

Total Cost $T_2 = \log_a n * bn$

" " $T_1 = \log_b n * an$

$a \geq b$

important

$\log a \geq \log b$ (Assuming a & $b \geq 1$)

$\frac{1}{\log a} \leq \frac{1}{\log b} \Rightarrow \log_a n \leq \log_b n$ — ①

also $b \leq a$ — ②

Multiply ① & ② ;

$\log_a n * b \leq \log_b n * a$, Multiply 'n'

$\log_a n \, bn \leq \log_b n \, an$

$T_2 \leq T_1$

T_1 grows greater or equal to T_2 for large values of n.