Problem set 1:

Q.1. Let us consider a general function

$$T(n) = a\,T\left(\frac{n}{b}\right) + cn^k \;;\; T(1) = c.$$

where $a, b, c, k \in$ constants.

~~let~~ without loss of generality, we can assume n to be a perfect power of b, since we are analysing the function as it grows large. (Asymptotic Analysis).

$\therefore$ let $n = b^r$ for some $r \in \mathbb{N}$.

$$\therefore T(n) = a\,T\left(\frac{n}{b}\right) + cn^k$$

$$T\left(\frac{n}{b}\right) = a\,T\left(\frac{n}{b^2}\right) + c\left(\frac{n}{b}\right)^k \times a$$

$$T\left(\frac{n}{b^2}\right) = a\,T\left(\frac{n}{b^3}\right) + c\left(\frac{n}{b^2}\right)^k \times a^2$$

$$T\left(\frac{n}{b^3}\right) = a\,T\left(\frac{n}{b^4}\right) + c\left(\frac{n}{b^3}\right)^k \times a^3$$

$$\vdots \quad \vdots \quad \vdots$$

$$+\; T\left(\frac{n}{b^{r-1}}\right) = a\,T\left(\frac{n}{b^r}\right) + c\left(\frac{n}{b^{r-1}}\right)^k \times a^{r-1}$$

$$T(n) = a^r T\left(\frac{n}{b^r}\right) + cn^k \sum_{i=0}^{r-1} \left|\frac{a}{b^k}\right|^i$$

Case 1: $a > b^k$

$$T(n) = a^r \times T(1) + cn^k \sum_{i=0}^{r-1} \left(\frac{a}{b^k}\right)^i$$

$$T(n) = ca^r + cn^k \times \frac{\left(\left(\frac{a}{b^k}\right)^r - 1\right)}{\frac{a}{b^k} - 1}.$$

Now, $a^r = a^{\log_b n} = n^{\log_b a}$.

and $b^r = n \Rightarrow b^{rk} = n^k$

$$T(n) = cn^{\log_b a} + c \times \frac{(a^r - b^{rk})}{\frac{a}{b^k} - 1}$$

$$T(n) = cn^{\log_b a} + \left(\frac{c}{\frac{a}{b^k} - 1}\right) \times n^{\log_b a} - \frac{c}{\frac{a}{b^k} - 1} \times n^k.$$

$$T(n) = \left(c + \frac{c}{\frac{a}{b^k} - 1}\right) n^{\log_b a} - \left(\frac{c}{\frac{a}{b^k} - 1}\right) \times n^k.$$

$$T(n) = c_1 n^{\log_b a} + c_2 n^k.$$

where $c_1 = c + \frac{c}{\frac{a}{b^k} - 1}$ and $c_2 = \frac{c}{1 - \frac{a}{b^k}}$

Now, if $a > b^k \Rightarrow \log_b a > k.$

$\Rightarrow n^{\log_b a} \geq n^k$

$\therefore T(n) = \Theta(n^{\log_b a})$

where $\Theta(\cdot)$ is the big-oh notation, and $f(n) = \Theta(n^r)$ means $f(n)$ grows no faster than n^r.

Case 2: $a < b^k$.

then $n^{\log_b a} < n^k$

$\therefore T(n) = \Theta(n^k)$

Case 3: $a = b^k$

Then $\sum_{i=0}^{r-1} \left(\frac{a}{b^k}\right)^i = \sum_{i=0}^{r-1} 1 = r.$

$\therefore T(n) = c\,a^r + cn^k \times (r)$

$= c\,n^{\log_b a} + c\,n^k \log_b n.$

$= cn^k + cn^k \log_b n.$

$= \Theta(n^k \log n).$

$$\therefore T(n) = \begin{cases} \Theta(n^k) & \text{if } a < b^k \\ \Theta(n^k \log n) & \text{if } a = b^k \\ \Theta(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$

This is Master Theorem.

Now, $T_1(n) = a\,T_1\left(\frac{n}{b}\right) + bn$. $(a > b)$

we have,

$$T_1(n) = \Theta(n^{\log_b a}).$$

and $T_2(n) = b\,T_1\left(\frac{n}{a}\right) + an$ $(a > b)$

$$T_2(n) = \Theta(n)$$

by applying Master Theorem.

$\therefore$ T_1 grows than T_2.

Q.2 Lemma: Let H be a sub-graph of G with more than two vertices, such that $\forall\, v \in V(H)$, $\deg(v) > 1$, then the graph G contains a Cycles.

Proof: let us assume not. Then we have have Graph G with sub-graph H with all vertices having $\because \deg(V) > 1$.

∴ By choosing a vertex v from $V(H)$, sufficiently long walk in $V(H)$, we would end up back at same vertex v and going for an arbitrary (self-loops not allowed; simple walk).

This means there is a cycle in H and ∴ in G, which is a contradiction.

∴ Lemma is true.

Algorithm for detecting cycle in an undirected Graph:

Cycle-Detect (G):

- Initialize array Degree[v] $\forall v \in V(G)$.

while ∃ a vertex that is not pushed into the DS (Data Structure)

$U \leftarrow$ set of vertices w/ degree < 2

If U is empty then return "Cycle Exist"

For all $v \in N(U)$:

Degree[v] = Degree[v] − 1

DS.append(U).

Complexity : $\Theta(m+n)$.

$m \rightarrow$ 'edges $|E|$

$n \rightarrow$ $|V|$. vertices.

Q.3 Let n_0 be the number of vertices with no child (leaves), n_2 be the number of vertices with 2 children and n_1 be number of vertices with 1 child.

∴ Since there are only three kinds of nodes, the sum of all three kinds should sum up to the total number of nodes. Let us say that is n.

∴ We have,

$$n = n_0 + n_1 + n_2 . \quad \rightarrow (i)$$

Now, we can calculate the total number of nodes by counting the children of every node.

∴ Total nodes = 2 × nodes having 2 children
+ 1 × nodes having 1 child
+ 1 (accounting for root).

$$\therefore \quad n = 2 \times n_2 + 1 \times n_1 \quad \rightarrow (ii)$$

combining (i) & (ii), we have

$$2n_2 + n_1 + 1 = n_0 + n_1 + n_2.$$

$$\Rightarrow \quad n_2 = n_0 - 1.$$

or

"In Any Binary Tree the number of nodes with two children is exactly one less than the number of leaves."

□

Q.4 Theorem: Between Any Two Vertices u and v of Tree T, there exist a unique path.

Proof: Assume not for the sake of contradiction.

Then there exist another path from u to v, which combined with the reverse of the first path would form a cycle, which is a contradiction since we assumed T is a Tree.

∴ A tree has unique path between any two vertices.

Now, let $(u, v) \in E(G)$ be an edge that is not in T.

In such a case:

→ in the DFS-tree, one of u or v should be the ancestor of the other

→ in the BFS-tree, u and v can differ by only one level.

Since we have DFST = BFST and both of the above points are true, (u, v) is also present in the Tree T, which is a contradiction.

The converse ⇒ When G is tree, DFST=BFST also holds due to Theorem-1. □

Q.5 Let us assume not for the sake of contradiction.

Then we have Graph G which is disconnected and has $deg(v) \geq n/2$ $\forall v \in V(G)$.

The graph could have many connected components, but let us assume it consists of only 2 component as this would cover other cases with more than 2 connected component.

Let $G = G_1 \cup G_2$ be the two ~~disjoint~~ disjoint component of G.

Let $n_1 = |V(G_1)|$ and $n_2 = |V(G_2)|$

$\Rightarrow n_1 + n_2 = n = |V(G)|$. $\rightarrow$ (i)

$\therefore n_1 \leq n/2$ or $n_2 \leq n/2$.

Without loss of generality, let $n_1 \leq n/2$.

$\Rightarrow$ G_1 has at most $n/2$ vertices.

$\Rightarrow$ Any $v \in V(G_1)$ has $\deg(v) \leq n/2 - 1$.

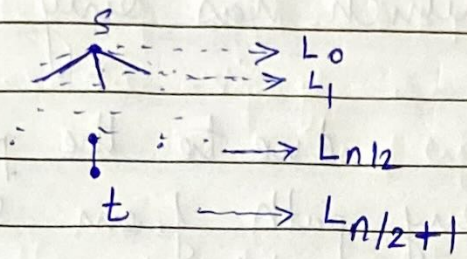
$\Rightarrow$ But we know all $v \in V(G) \supset V(G_1)$ have $\deg(v) \geq n/2$.

$\Rightarrow$ This is a contradiction.

$\therefore$ The Graph G is connected.

$\rightarrow$ Cases of more than 2 connected component in the proof can be handled replacing $n/2$ by n/r in eqn. (i), where r = number of connected component (>2)

Q.6 Consider a BFS tree rooted at s.



Given, t is at a distance that is strictly greater than $n/2$. $[|V| = n]$

∴ and since vertices at level L_y are are at a distance x from s,

∴ t must at L_x, where $x \geq n/2 + 1$.

Let us assume, without loss of generality that $x = n/2 + 1$.

∴ we have layers $L_1, L_2, L_3, \ldots, L_{n/2}$ between s and t.

And between these layers, there can be at most $n-2$ nodes (excluding s and t).

Now, each of these layers cannot have 2 or more vertices. since

$$\underbrace{2 + 2 + \ldots + 2}_{n/2 \text{ times}} = n > n-2.$$

∴ There exist a layer between 0 to $n/2+$

which has only one node, say L_K.

Now, due to the property of BFS tree every path from s to t must travel through at least one node of each layer from s to t.

∴ Every path goes through the only node present in layer L_K. (say v).

⇒ Removing v destroys all paths from s to t present in graph G.

Algorithm : i) Run a BFS at vertex s.

ii) Maintain a List L[i] for each node present at distance i from s in the BFS tree.

iii) Find a Layer L[i] which has only one node present in it, say v. $(1 \le j \le n/2)$.

iv) Output v.

The complexity of the algorithm is same as the complexity of BFS, which is

$O(m+n)$, $m = |E(G)|$, $n = |V(G)|$.

References :

Q.1 A variant of Master's Theorem taken from "https://randerson112358.medium.com/master-theorem-909f52d4364".
Proof was completely done by me.

Q.2 Idea ~~concise~~ of lemma conceived by me, inspired from lecture slide 5 page 3. (Topological sort)

Q.3 I remember the solution to this question as I came across it while preparing from for my LEEE exam. one of previous year's Gate Questions from Discrete Mathematics section.

Q.4 1) ~~Definition~~ Definition of Tree from book "Discrete Mathematics and Its Application 7th Edition" by Kenneth H. Rosen. page no. 746.

2) "https://cse.iitkgp.ac.in/~agupta/Algo-1/Tutorial-6-Sol.pdf."

3) "~~https~~ "https://stackoverflow.com/questions/3688083/equivalence-of-a-graph-and-a-bfs-and-dfs-tree".

Q.5 Discussed with Arnav Negi (Roll no. 2021101112)

Q.6 1) "https://math.stackexchange.com/questions/2144357/proof-that-deleting-a-certain-node-in-a-graph-where-two-nodes-have-a-path-of-len".

2) "https://www.albany.edu/~ravi/pdfs-csi604/sol_hw7.pdf"