Y. Shivananda B210603CS CS04

Q1) Prove that clique is NP

Ans) Problem Definition:

A clique in an undirected graph $G=(V,E)$ is a subset of vertices $S \subseteq V$ such that every pair of vertices in S is connected by an edge.

Input:

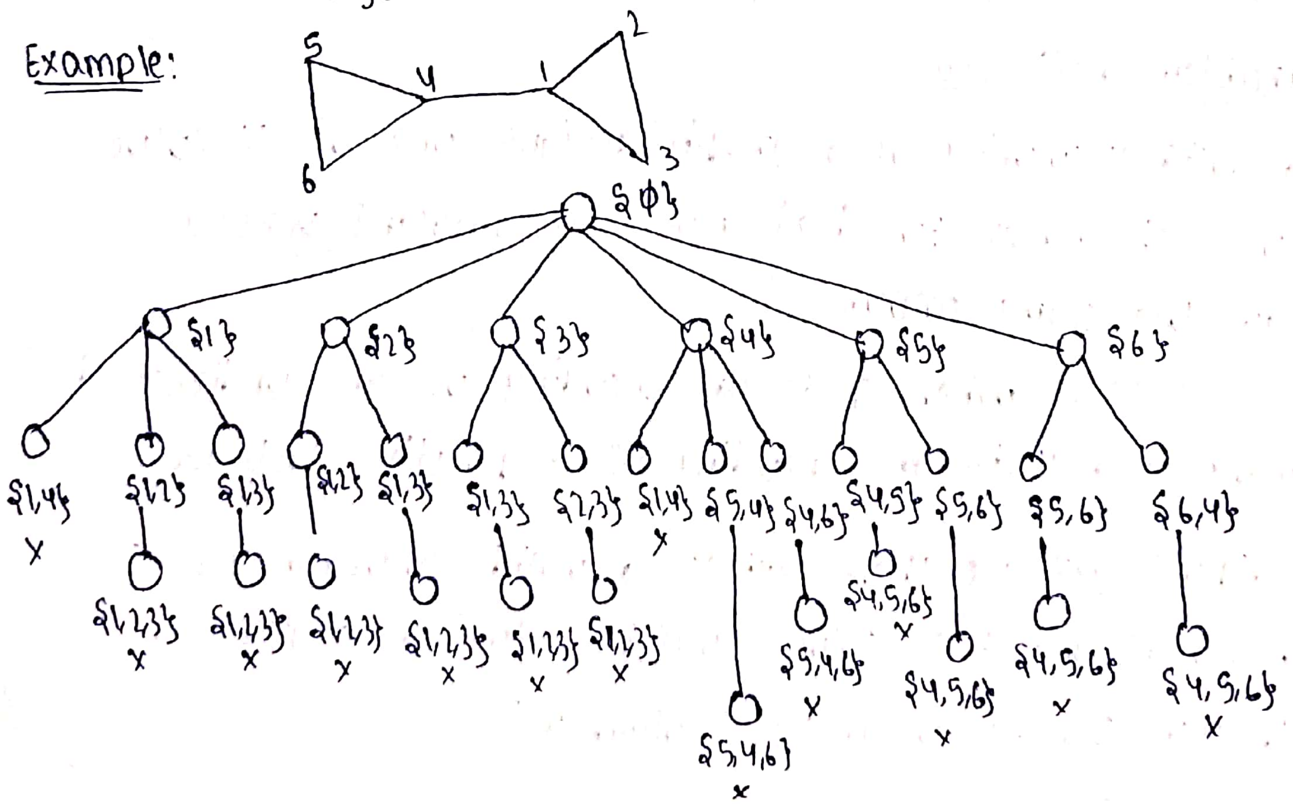
$\langle G, K \rangle$ - Encoding of an undirected graph G and a positive integer K.

Output:

"yes" If there is a vertex subset of size K or more vertices in G where every pair is adjacent. "No" otherwise.

We can design a Non-deterministic Turing machine that executes this algorithm.

Example:



x → Indicating Halting

Initially, we start with an empty solution set. At each branching step, we add a distinct vertex to every solution set under consideration. Note that, A vertex is added only if it is adjacent to all vertices already present in the solution set. If there are no more vertices can be added, we halt.

Every Adjacency check → $O(|V| \times |E|)$ time

No. of nodes in a Branch → $O(|V|)$ nodes

Thus, this Non-deterministic Turing machine can determine whether $\langle G, k \rangle \in$ CLIQUE in $O(|V|^2)$ [polynomial time]

Therefore, we can say that clique is NP.

Q) Prove that colouring is NP.

Ans) <u>Problem Definition:</u>

A valid K-coloring of a graph assigns one of K colors to each vertex so that no two adjacent vertices have the same colour.

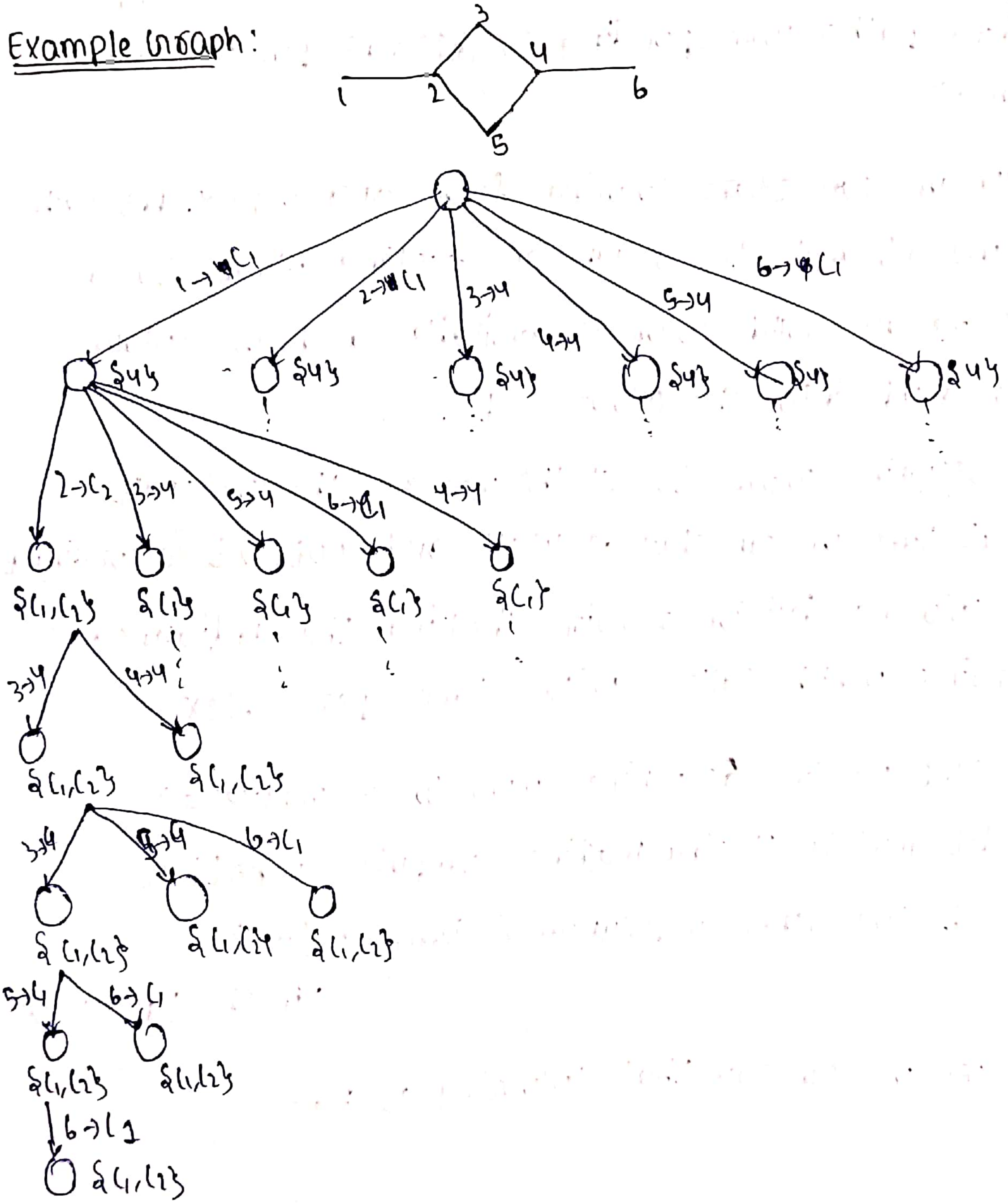
<u>Input:</u> $\langle G, K \rangle$ where G is an Encoding of an undirected graph and K is an Encoding of a positive Integer.

<u>Output:</u> "yes" If there exists a valid coloring of G using at most K colors. [valid coloring refers as $\forall (u,v) \in G.E: \text{Colour}(u) \neq \text{Colour}(v)$]

"No" otherwise.

We can design a Non-deterministic Turing Machine that executes the following algorithm:- {let E = G.E and V = G.V}

Example Graph:



Halts! [All vertices are assigned colors]

Note: The Dotted Lines are also expanded through all the possible ways.

In the operation of this Non-deterministic Turing Machine (NTM), each vertex is assigned a colour, with multiple branching paths corresponding to different possible colour assignments.

Initially, we can start coloring from any vertex. At each step, we attempt to assign an existing colour to a vertex. If no valid color is available, a New colour is Introduced into the colour set C.

After assigning colours to all the vertices, the TM Halts. At that time, we can check if $|C| \leq K$ and output accordingly. Each coloring decision requires checking the Adjacency constraints, which takes $O(|V|)$ time per vertex.

Each step involves assigning colors to $|V|$ vertices $\Rightarrow$ Non-deterministic Turing Machine compute whether $\langle G, K \rangle \in$ COLOURING in polynomial time $O(|V|^2)$

↳ in the I/P length

Therefore, we can say that colouring is NP.