**Problem Name:** Hamiltonian Path

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

**Input Format:**

**Output Format:**

**Constraints:**

**Examples:**

**Brute force Solution:**

**Explanation:**

The simplest approach to solve the given problem is to [generate all the possible permutations](https://www.geeksforgeeks.org/write-a-c-program-to-print-all-permutations-of-a-given-string/) of**N** vertices. For each permutation, check if it is a valid Hamiltonian path by checking if there is an edge between adjacent vertices or not. If found to be true, then print **“Yes”**. Otherwise, print **“No”**.

**Code:**

**Time Complexity**: *O(N \* N!)*

**Space Complexity: O(1)**

**Optimized Solution:**

Explanation:

Depth first search and backtracking can also help to check whether a Hamiltonian path exists in a graph or not. Simply apply depth first search starting from every vertex v and do labeling of all the vertices. All the vertices are labelled as either "IN STACK" or "NOT IN STACK". A vertex is labelled "IN STACK" if it is visited but some of its adjacent vertices are not yet visited and is labelled "NOT IN STACK" if it is not visited.  
If at any instant the number of vertices with label "IN STACK" is equal to the total number of vertices in the graph then a Hamiltonian Path exists in the graph.

**Code:**

Time Complexity: O(N!)

**Space Complexity:**