Experiment **#15**		Student ID	
Date		Student Name	[@KLWKS_BOT THANOS]

Experiment Title: Implementation of basic programs on Non-Deterministic Algorithms - I.

Aim/Objective: To understand the concept and implementation of Basic programs on Non-Deterministic Algorithms.

Description:

The students will able to understand and implement programs on Non-Deterministic Algorithms.

Pre-Requisites:

Knowledge: Non-Deterministic Algorithms in C/C++/Java/Python

Tools: Code Blocks/Eclipse IDE

Pre-Lab:

Read the following conversation

Jaya: Travelling salesman problem is a NP hard problem. Hema: I do not think so

Jaya: No, I am so sure that Travelling Salesman problem is a NP hard problem. Hema: …!!

You are Jaya's friend. Help her prove her statement.

Procedure:

1. **Definition of TSP:**
   - TSP asks whether there exists a Hamiltonian cycle in a weighted graph such that the total weight is less than or equal to a given value k.
2. **Belongs to NP:**
   - Given a proposed solution (a path), it can be verified in polynomial time whether:
     - The path visits every vertex exactly once.
     - The total cost is less than or equal to k.
3. **Reduction from Hamiltonian Cycle:**
   - The **Hamiltonian Cycle Problem** (HCP) is a known NP-complete problem.
   - HCP can be reduced to TSP:
     - Assign weights of 1 to edges in the Hamiltonian cycle.
     - Assign very large weights (e.g., infinity) to all other edges.
     - Solving the TSP in this case also solves HCP.

4. **Conclusion**:
   - Since TSP can verify solutions in polynomial time and a known NP-complete problem (HCP) reduces to it, TSP is NP-hard.

Jaya is correct.

In-Lab:

Raju prepares for the examination, but he got stuck into a concept called "NP-HARD AND "NP-COMPLETE PROBLEMS" on Nondeterministic Algorithms. So, help Raju to score good marks. Help him to define the Nondeterministic algorithms by sorting an array.

Procedure/Program:

Nondeterministic Algorithm - Explanation Using Array Sorting

A **nondeterministic algorithm** is a conceptual model where the algorithm can "guess" solutions instantly and verify them efficiently. Sorting an array using such an algorithm involves:

1. **Guessing**: The algorithm guesses a permutation of the array.
2. **Verification**: It checks if the guessed permutation is sorted in $O(n)$ time.

Though nondeterministic algorithms are theoretical, they highlight the concept of solving problems efficiently if "perfect guesses" were possible.

- **Data and Results:**

Data

Array of integers to be sorted using nondeterministic algorithm.

Result

Guesses a permutation and verifies if it's sorted efficiently.

- **Analysis and Inferences:**

Analysis

Verification is polynomial, guessing theoretically bypasses exhaustive search process.

Inferences

Nondeterministic sorting highlights efficiency with hypothetical perfect guessing.

Post-Lab:

Hema: Hamiltonian Path is NP-Complete.

Jaya: Well, prove that!

Hema: I will prove and let you know.

Help Hema to try and prove that Hamilton Path is NP-Complete

Procedure/Program:

1. **Hamiltonian Path in NP**:
   - A given path can be verified in polynomial time to check if it visits all vertices exactly once.
2. **Reduction from Hamiltonian Circuit (HC)**:
   - HC asks for a cycle visiting each vertex exactly once.
   - Transform HC graph G to a new graph G' by splitting a vertex v into v_1 and v_2 with an edge $v_1 \rightarrow v_2$.
   - HC in G corresponds to HP in G'.
3. **Conclusion**:
   - HP is NP (verification) and NP-hard (reduction).
   - Thus, **Hamiltonian Path is NP-Complete**.

- **Data and Results:**

Data

Hamiltonian Path verification can be done in polynomial time efficiently.

Result

Hamiltonian Path problem is proven to be NP-Complete.

- **Analysis and Inferences:**

Analysis

Reduction from Hamiltonian Circuit ensures NP-hardness of Hamiltonian Path.

Inferences

Hamiltonian Path combines verification and reduction for NP-Completeness proof.

- **Sample VIVA-VOCE Questions:**

1. How does a non-deterministic algorithm differ from a deterministic algorithm?

- Deterministic: Executes a single sequence of steps.
- Non-deterministic: Explores multiple possibilities simultaneously.

2. Is the clique decision problem in the class NP?

- Yes, verifying a solution (a clique of size k) is possible in polynomial time.

3. What is the NP complexity class? What is the relationship between NP and P complexity classes?

- NP: Problems verifiable in polynomial time.
- P $\subseteq$ NP; unknown if $P = NP$.

4. What is the power of non-deterministic algorithms? How are they related to the concept of polynomial-time verification?

- Non-deterministic algorithms solve problems by guessing and verifying solutions in polynomial time.

5. Give some applications of non-deterministic algorithms.

- Applications: Traveling Salesman Problem, graph coloring, scheduling, and optimization problems.

Evaluator Remark (if Any):	**Marks Secured___ out of 50**
	Signature of the Evaluator with Date

Evaluator MUST ask Viva-voce prior to signing and posting marks for each experiment.