Date of the Session:___/___/_____ **Time of the Session:______to_______**

EX – 12 Working with NP Hard and NP Complete Problems**.**

Prerequisites:

- Basics of Data Structures and C Programming.
- Basic knowledge about arrays and graphs.

Pre-Lab:

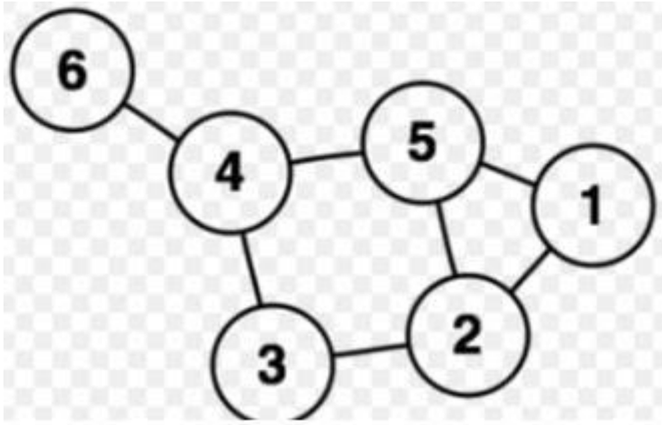
1) Read the following conversation
Jaya: Travelling salesman problem is a NP hard problem.
Hema: I do not think so
Jaya: No, I am so sure that Travelling Salesman problem is a NP hard problem.
Hema: …!!
You are Jaya's friend. Help her prove her statement.

- **Definition**: A problem is **NP-hard** if solving it efficiently would allow us to solve all problems in NP efficiently.
- **Reduction**: The **Hamiltonian Cycle Problem** (which is NP-complete) can be reduced to the **Travelling Salesman Problem (TSP)** in polynomial time.
- **Implication**: Since Hamiltonian Cycle is NP-hard, and TSP is at least as hard as it, TSP is also **NP-hard**.

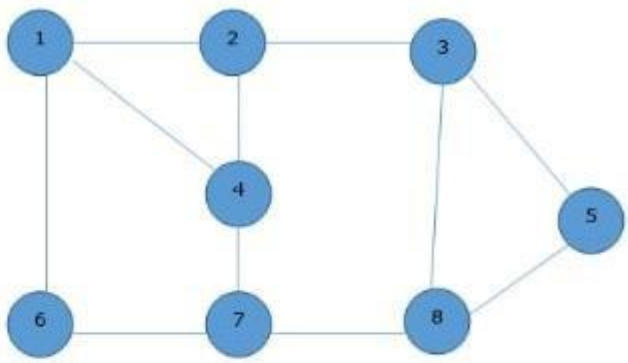
[@KLWKS_BOT] THANOS

2) Consider the two graphs and find out the node cover for the each of the graphs and specify the maximum node cover.

a)



b)



Graph 1 Analysis:

The first graph consists of 6 nodes. To find a **minimum vertex cover**, we aim to cover all edges with the fewest nodes.

- **One possible minimum node cover:** {2, 3, 4, 5}
- **Maximum node cover:** All 6 nodes.

Graph 2 Analysis:

The second graph consists of 8 nodes and has a more structured layout.

- **One possible minimum node cover:** {2, 3, 4, 5, 6, 8}
- **Maximum node cover:** All 8 nodes.

In both graphs, the **maximum node cover** is the entire set of nodes.

[@KLWKS_BOT] THANOS

In-Lab :

1) 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.

Source code:

A **Nondeterministic Algorithm** consists of two phases:

1. **Guessing (Nondeterministic Phase):** Generates a potential solution.
2. **Verification (Deterministic Phase):** Checks if the guessed solution is correct in polynomial time.

Sorting an Array using a Nondeterministic Algorithm:

1. **Guess:** Randomly generate a permutation of the array.
2. **Verify:** Check if the guessed permutation is sorted in **O(n)** time.

2) Karthik and Bavya are trying to implement a program that helps a transportation company use its container to maximum efficiency, as part of their CS project.
Few objects are more valuable than others. Each object has a certain weight. The transportation company wants to fill the container so that the value of the container (sum of value of objects in the container) is maximum, while total weight of the objects does not exceed the container's capacity.
As the outcome is not fixed, this is a non-deterministic problem. This is a knapsack problem as we must maximize value within the given weight limit.
So, to understand the problem well and implement it, help them in finding non-deterministic knapsack algorithm.

Source code:

```
#include <stdio.h>

int max(int a, int b) {
   return (a > b) ? a : b;
}

int knapsack(int i, int weight, int value, int W, int weights[], int values[], int n) {
   if (weight > W) return 0;
   if (i == n) return value;

   int include = knapsack(i + 1, weight + weights[i], value + values[i], W, weights, values, n);
   int exclude = knapsack(i + 1, weight, value, W, weights, values, n);

   return max(include, exclude);
}

int main() {
   int weights[] = {2, 3, 4, 5};
   int values[] = {3, 4, 5, 6};
   int W = 5;
   int n = sizeof(weights) / sizeof(weights[0]);

   printf("Maximum Value: %d\n", knapsack(0, 0, 0, W, weights, values, n));
   return 0;
}
```

[@KLWKS_BOT] THANOS

Post-Lab:

1) 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

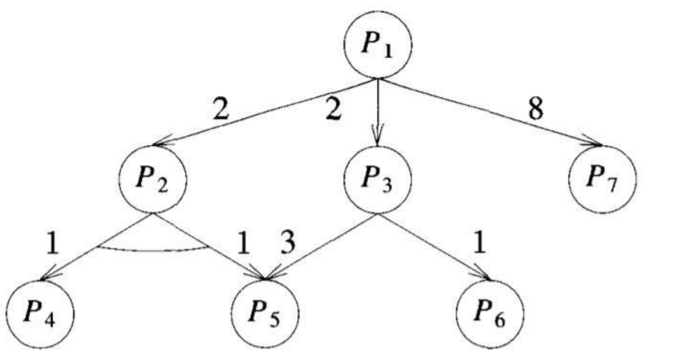
Source code:

Proof that Hamiltonian Path is NP-Complete

1. **In NP**: A given path can be verified in polynomial time.
2. **NP-Hard**: Reduce from **Hamiltonian Cycle** (HC, known NP-Complete).
   - Given a graph G with HC, pick a vertex v, split it into v_1, v_2.
   - If G has HC, the modified graph has a Hamiltonian Path from v_1 to v_2.
3. Since Hamiltonian Path is **in NP** and **NP-Hard**, it is **NP-Complete**.

[@KLWKS_BOT] THANOS

2) Hemanth was unable to answer the following question in exam. Here is the question, help Hemanth to find P1 solution. Find the total cost to find the P1 solution



Source code:

Paths and Costs:

- **P4 → P2 → P1** = 1 + 2 = **3**
- **P5 → P3 → P1** = 3 + 2 = **5**
- **P6 → P3 → P1** = 1 + 2 = **3**
- **P7 → P1** = 8

Minimum Cost Path:

The shortest path to **P1** is from **P4 → P2 → P1** or **P6 → P3 → P1**, both having a total cost of **3**.

Answer:

The minimum total cost to reach P1 is 3.

Comments of the Evaluators (if Any)	Evaluator's Observation
	Marks Secured:_______out of [50]. Signature of the Evaluator Date of Evaluation: