Assignment: P vs NP - Problem Selection and Solution

Objective

Demonstrate your understanding of computational complexity by identifying, formulating, and solving real-world problems from both P and NP complexity classes.

Assignment Requirements

Part 1: Choose and Solve a P Problem

1. **Select** any problem you know to be in class P (solvable in polynomial time) from the acceptable list below
2. **Create** a real-world scenario/application for your chosen problem
3. **Generate** a small problem instance with 10-15 items/nodes/elements
4. **Solve** the instance using an appropriate algorithm
5. **Provide** step-by-step solution showing your work
6. **Analyze** the time complexity and explain why this problem belongs to class P

Part 2: Choose and Solve an NP Problem

1. **Select** any problem you know to be NP-complete or NP-hard from the acceptable list below
2. **Create** a real-world scenario/application for your chosen problem
3. **Generate** a small problem instance with 10-15 items/nodes/elements
4. **Solve** the instance optimally (you may use any method: brute force, dynamic programming, branch and bound, etc.)
5. **Provide** the optimal solution with clear justification that it is indeed optimal
6. **Analyze** why this problem is computationally harder and explain its NP classification

---

Acceptable Problems

P Problems (Choose ONE):

- Dijkstra's Shortest Path Algorithm
- Kruskal's or Prim's Minimum Spanning Tree
- Topological Sorting (DAG)
- Maximum Flow (Ford-Fulkerson, Edmonds-Karp)
- Convex Hull (Graham Scan, Jarvis March)
- All-Pairs Shortest Path (Floyd-Warshall)
- Strongly Connected Components (Tarjan's, Kosaraju's)
- Maximum Bipartite Matching (Hungarian Algorithm)
- Longest Common Subsequence (Dynamic Programming)

- Binary Search Tree Operations (insertion, deletion with balancing)

Note: Your chosen P problem should demonstrate non-trivial algorithmic thinking beyond simple loops.

NP Problems (Choose ONE):

- Traveling Salesman Problem (TSP)
- 0/1 Knapsack Problem
- Graph Coloring (with ≥3 colors)
- Boolean Satisfiability (3-SAT or higher)
- Set Cover Problem
- Bin Packing Problem
- Job Shop Scheduling
- Vehicle Routing Problem
- Maximum Clique Problem (clique size ≥3)
- Hamiltonian Cycle/Path
- Vertex Cover Problem
- Partition Problem
- Subset Sum Problem

Note: You can choose other problems, but not general ones.