Mid Term Exam

1. **Write code** to create a **directed** graph with **100 nodes** where each node is numbered between **2** and **101**. There exists an edge from **node-A** to **node-B** if **node-B** is a multiple of **node-A.** **10**

   For example, the node **50** should have directed edges to node **50** and node **100.** (as both 50 and 100 are multiples of 50)

   Use **adjacency list** as your graph representation.

2. Can you draw the graph in **problem-1** for **10** nodes where all nodes are numbered from **2** to **11.**
   Will this graph be a **DAG** if there are a million nodes? Why or why not?
   10

3. Can you find the length of the **longest path** in **problem-1**?
   10
   What graph traversal would you use? Can it be solved with both **BFS** and **DFS**?
   Finally, **write code** to print the longest path.

4. Given a positive integer **n**, write a **recursive** function to print all combinations of numbers between **1** and **n** having sum **n**.
   10
   For example,

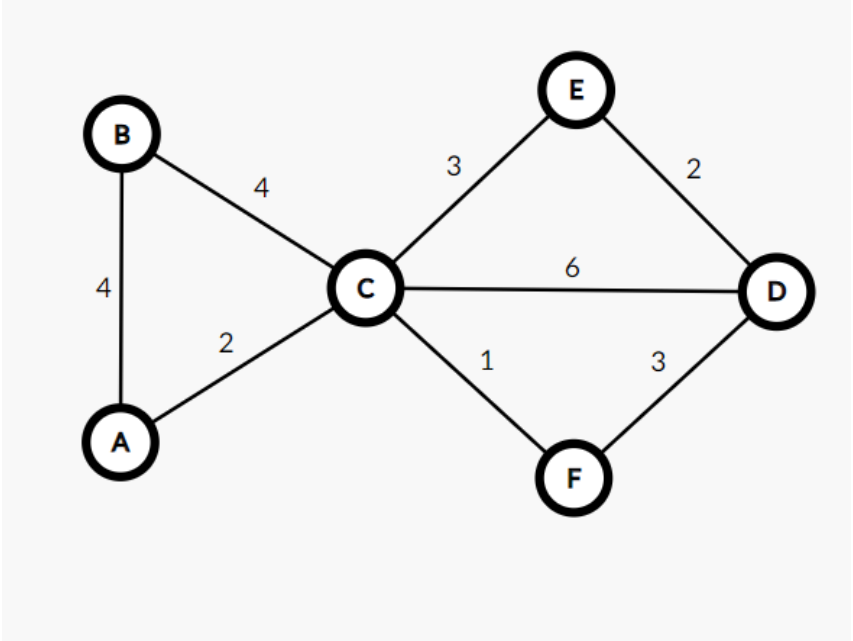
   For **n = 5**, the following combinations are possible:
   - `{ 5 }`
   - `{ 1, 4 }`
   - `{ 2, 3 }`
   - `{ 1, 1, 3 }`
   - `{ 1, 2, 2 }`
   - `{ 1, 1, 1, 2 }`
   - `{ 1, 1, 1, 1, 1 }`

5. **Write code** to solve the following problem: SPOJ.com - Problem ABCPATH
   10

6. **Write code** to solve the following problem: [Problem - 580C - Codeforces]
   10

7. **Write code** to solve the following problem: [Country Roads | LightOJ]
   10

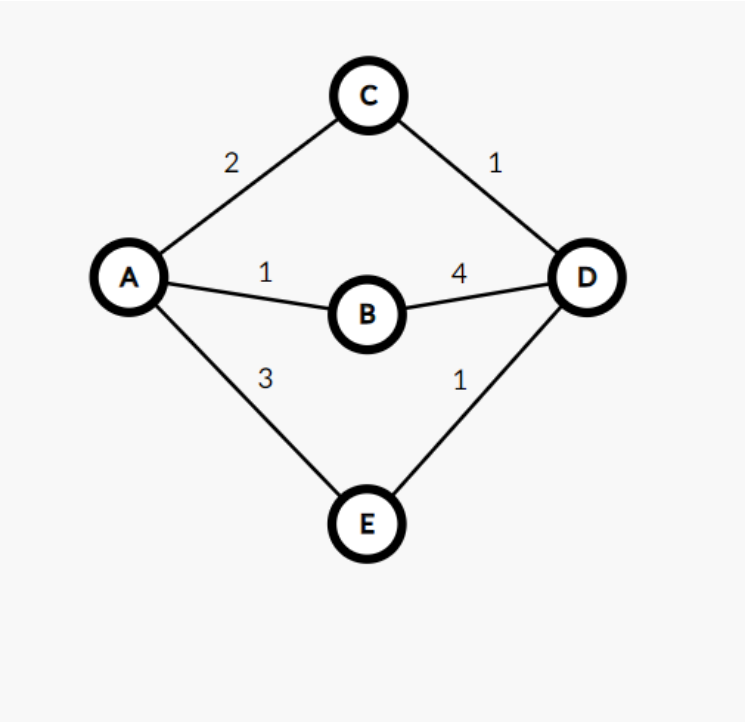
8. Simulate **Dijkstra**'s algorithm (the optimised version) on the following graph:
   Use **node-B** as source.
   15



9. We know that in a **weighted graph**, we can detect the **smallest distance** of a node from the source node using **Dijkstra's** algorithm. But can we detect the **2nd smallest distance** of a node from the source node?
   10

   For example, in the following graph the **smallest distance** from **node-A** to **node-D** is **3** if we take the path `A -> C -> D`

   But the **2nd smallest distance** from **node-A** to **node-D** is **4** if we take the path `A -> E -> D`
   How can you modify **Dijkstra's** algorithm to solve this problem? (code is not needed. You can write your idea or pseudocode)



10. Will **Dijkstra's** algorithm give correct output for the following graph?

5

Why or why not? The source node is **node-A** and the destination is **node-B.**