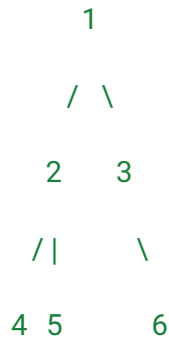


## 1. Dijkstra's Algorithm

Implement Dijkstra's algorithm to find the shortest path from a source vertex to all other vertices in a weighted graph. Ensure your solution handles edge cases such as disconnected graphs or negative weights.

**Input Example:**

Graph:



Edges and Weights:

1 -> 2 (4), 1 -> 3 (2), 2 -> 4 (7), 2 -> 5 (1), 3 -> 6 (5), 5 -> 6 (3)

Source: 1

**Output Example:**

Shortest Path from Source (1): {1: 0, 2: 4, 3: 2, 4: 11, 5: 5, 6: 8}

---

## 2. Rotate a Matrix

Rotate a given N x N matrix 90 degrees clockwise in-place.

**Input Example:**

```
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
```

```
]
```

**Output Example:**

```
matrix = [  
    [7, 4, 1],  
    [8, 5, 2],  
    [9, 6, 3]  
]
```

---

**3. Maximum Frequency Stack**

Design a stack-like data structure that pushes and pops elements based on their frequency. If two elements have the same frequency, the most recently pushed element is popped first.

**Input Example:**

Operations:

```
push(5), push(7), push(5), push(7), push(4), push(5)
```

```
pop(), pop(), pop(), pop()
```

**Output Example:**

Popped Elements:

```
5, 7, 5, 4
```

---

**4. Sliding Window Maximum**

For a given array and a number `k`, find the maximum for each and every contiguous subarray of size `k`.

**Input Example:**

```
arr = [1, 3, -1, -3, 5, 3, 6, 7], k = 3
```

**Output Example:**

```
[3, 3, 5, 5, 6, 7]
```

---

**5. Detect and Remove a Loop in a Linked List**

Detect and remove the loop in a linked list using Floyd's Cycle Detection algorithm.

**Input Example:**

Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> 3 (Loop back to 3)

**Output Example:**

Modified Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> NULL

---

**6. Find the Median of a Running Stream of Integers**

Design a data structure that efficiently supports the following operations:

- Add an integer to a running stream.
- Retrieve the median of all elements added so far.

**Input Example:**

Stream: [6, 10, 2, 8, 4, 12]

**Operations:**

```
add(6), add(10), getMedian(), add(2), add(8), getMedian(), add(4),  
add(12), getMedian()
```

**Output Example:**

After add(6): Median = 6

After add(10): Median = 8

After add(2): Median = 6

After add(8): Median = 7

After add(4): Median = 6

After add(12): Median = 7

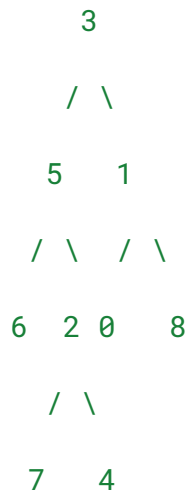
---

### 7. Lowest Common Ancestor

Find the lowest common ancestor of two nodes in a binary tree.

**Input Example:**

Binary Tree:



Nodes: 5, 1

**Output Example:**

LCA: 3

---

### 8. Topological Sorting

Given a directed acyclic graph (DAG), implement topological sorting using both DFS and Kahn's Algorithm.

**Input Example:**

Graph:

1 -> 2

1 -> 3

3 -> 4

2 -> 4

**Output Example:**

Topological Order: [1, 3, 2, 4] (or [1, 2, 3, 4] depending on the implementation)

---

**9. Count Distinct Subsequences**

Given a string, count the number of distinct subsequences (non-contiguous substrings) that can be formed from the string.

**Input Example:**

string = "abcbac"

**Output Example:**

Number of Distinct Subsequences: 32