### Algorithm: Floyd-Warshall Algorithm

Sir Cedric and Ember faced the Maze of Mirrors, a complex labyrinth with numerous paths reflecting light. To save Princess Elara, they needed to determine the shortest path between all pairs of mirrors.

#### Initialize Data Structures:

* Sir Cedric used an enchanted crystal (2D array) to keep track of the shortest distances between mirrors.

#### Reflect the Light:

* Sir Cedric recorded the initial paths between the mirrors.
* He adjusted the paths, finding shorter routes through other mirrors.

#### Retrieve the Result:

* The crystal showed the shortest paths between all pairs of mirrors.

#### Implementation:

| **def** floyd\_warshall(mirrors: List[List[int]]) -> List[List[int]]:  n = len(mirrors)  dist = [[float('inf')] \* n **for** \_ **in** range(n)]  **for** i **in** range(n):  **for** j **in** range(n):  **if** i == j:  dist[i][j] = 0  **elif** mirrors[i][j]:  dist[i][j] = mirrors[i][j]  **for** k **in** range(n):  **for** i **in** range(n):  **for** j **in** range(n):  dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])  **return** dist  *# Example usage:*  mirrors = [  [0, 3, float('inf'), 7],  [8, 0, 2, float('inf')],  [5, float('inf'), 0, 1],  [2, float('inf'), float('inf'), 0]  ]  print(floyd\_warshall(mirrors)) *# Output: Shortest path matrix* |
| --- |

#### Explanation:

Initialize:

* dist: An enchanted crystal to track the shortest paths between mirrors.

Reflect the Light:

* Sir Cedric recorded and adjusted paths to find shorter routes through other mirrors.

Retrieve the Result:

### The crystal showed the shortest paths between all pairs of mirrors.