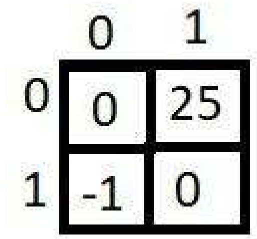
# Floyd Warshall

Difficulty: **Medium** Accuracy: **32.89%** Submissions: **160K+** Points: **4**

The problem is to find the shortest distances between every pair of vertices in a given edge-weighted **directed** graph. The graph is represented as an adjacency matrix. **mat[i][j]** denotes the weight of the edge from i to j. If **mat[i][j] = -1,** it means there is no edge from i to j.
Note: Modify the distances for every pair in place**.**

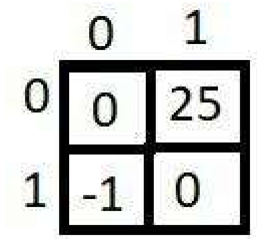
**Examples :**

**Input:** mat = [[0, 25], [-1, 0]]



|   | 0 | 1 |
|---|---|---|
| 0 | 0 | 25 |
| 1 | -1 | 0 |

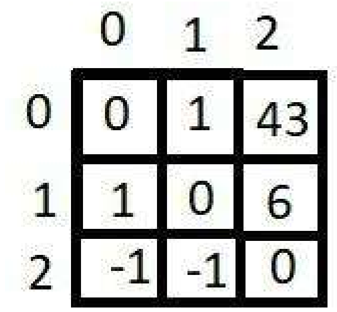
**Output:** [[0, 25], [-1, 0]]



|   | 0 | 1 |
|---|---|---|
| 0 | 0 | 25 |
| 1 | -1 | 0 |

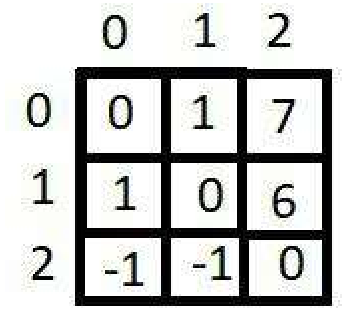
**Explanation:** The shortest distance between every pair is already given(if it exists).

**Input:** mat = [[0, 1, 43],[1, 0, 6], [-1, -1, 0]]



|   | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 1 | 43 |
| 1 | 1 | 0 | 6 |
| 2 | -1 | -1 | 0 |

**Output:** [[0, 1, 7], [1, 0, 6], [-1, -1, 0]]



|   | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 1 | 7 |
| 1 | 1 | 0 | 6 |
| 2 | -1 | -1 | 0 |

**Explanation:** We can reach 2 from 0 as 0->1->2 and the cost will be 1+6=7 which is less than 43.

**Constraints:**

1 <= mat.size() <= 100

-1 <= mat[ i ][ j ] <= 1000

Try more examples

**Company Tags**

## Topic Tags

## Related Articles

## Expected Complexities