a)To represent the graph in Figure 1, I would use an adjacency matrix as shown below

```
# The symmetric distance matrix for 7 cities (labeled 1-7)
# Row i and column j represent the distance from City (i+1) to City (j+1)
# A value of 0 indicates no direct connection between cities
dist = [
    [0, 12, 10,  0,  0,  0, 12],
    [12, 0,  8, 12,  0,  0,  0],
    [10, 8,  0, 11,  3,  0,  9],
    [0, 12, 11,  0, 11, 10,  0],
    [0,  0,  3, 11,  0,  6,  7],
    [0,  0,  0, 10,  6,  0,  9],
    [12, 0,  9,  0,  7,  9,  0]
]
```

a) Justification:

Simplicity for symmetrical distances: Since the problem involves symmetric distances (the distance from city A to city B is the same as from B to A), the matrix is symmetric across its diagonal, making it easy to verify data integrity and simpler to understand.

Constant-time access: The distance between any two cities can be retrieved in O(1) time, which is crucial for the repeated distance lookups required in the TSP algorithm. When calculating potential paths, we need to perform many distance lookups, and the immediate access provided by an adjacency matrix makes these operations very efficient.