Lab Goal: This lab was designed to teach you how to implement a graph with an adjacency list and matrix.

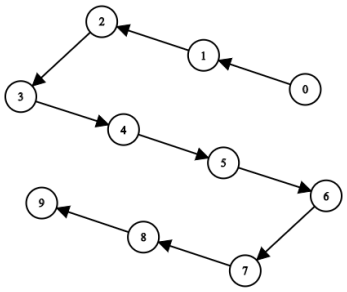
Lab Description : Finish the method degreeSequence from the abstract class Graph. Write the method getDistance2 for both GraphAdjList and GraphAdjMat (the order of the list isn't important). All the test cases have been provided for you. In this graph implementation nodes are represented by integers starting from 0 to n.

Files Needed ::
Graph.java
GraphAdjList.java
GraphAdjMat.java

Sample Output:

Straight line (0->1->2->3->...)

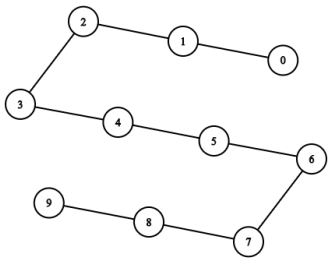
```
Graph with 10 vertices and 9 edges.
Degree sequence: [2, 2, 2, 2, 2, 2, 2, 2, 1, 1].
Adjacency list (size 10+9 = 19 integers):
        0: 1,
        1: 2,
        2: 3,
        3: 4,
        4: 5,
        5: 6,
        6: 7,
        7: 8,
        8: 9,
        9:
```



```
Graph with 10 vertices and 9 edges.
Degree sequence: [2, 2, 2, 2, 2, 2, 2, 2, 1, 1].
Adjacency matrix (size 10x10 = 100 integers):
  0: 0, 1, 0, 0, 0, 0, 0, 0, 0, 0
  1: 0, 0, 1, 0, 0, 0, 0, 0, 0, 0
  2: 0, 0, 0, 1, 0, 0, 0, 0, 0, 0
  3: 0, 0, 0, 0, 1, 0, 0, 0, 0, 0
  4: 0, 0, 0, 0, 0, 1, 0, 0, 0, 0
  5: 0, 0, 0, 0, 0, 0, 1, 0, 0, 0
  6: 0, 0, 0, 0, 0, 0, 0, 1, 0, 0
  7: 0, 0, 0, 0, 0, 0, 0, 0, 1, 0
  8: 0, 0, 0, 0, 0, 0, 0, 0, 0, 1
  9: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Undirected straight line (0<->1<->2<->3<->...)

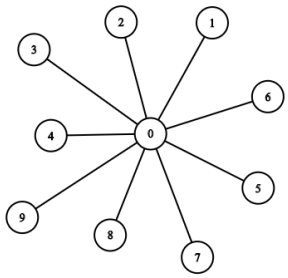
```
Graph with 10 vertices and 18 edges.
Degree sequence: [2, 2, 2, 2, 2, 2, 2, 2, 1, 1].
Adjacency list (size 10+18 = 28 integers):
        0: 1,
        1: 2, 0,
        2: 3, 1,
        3: 4, 2,
        4: 5, 3,
        5: 6, 4,
        6: 7, 5,
        7: 8, 6,
        8: 9, 7,
        9: 8,
```



```
Graph with 10 vertices and 18 edges.
Degree sequence: [2, 2, 2, 2, 2, 2, 2, 2, 1, 1].
Adjacency matrix (size 10x10 = 100 integers):
  0: 0, 1, 0, 0, 0, 0, 0, 0, 0, 0
  1: 1, 0, 1, 0, 0, 0, 0, 0, 0, 0
  2: 0, 1, 0, 1, 0, 0, 0, 0, 0, 0
  3: 0, 0, 1, 0, 1, 0, 0, 0, 0, 0
  4: 0, 0, 0, 1, 0, 1, 0, 0, 0, 0
  5: 0, 0, 0, 0, 1, 0, 1, 0, 0, 0
  6: 0, 0, 0, 0, 0, 1, 0, 1, 0, 0
  7: 0, 0, 0, 0, 0, 0, 1, 0, 1, 0
  8: 0, 0, 0, 0, 0, 0, 0, 1, 0, 1
  9: 0, 0, 0, 0, 0, 0, 0, 0, 1, 0
```

Star graph - 0 is connected in both directions to all nodes except itself (starting at 0)

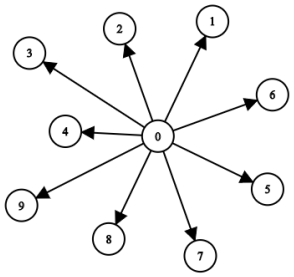
```
Graph with 10 vertices and 18 edges.
Degree sequence: [9, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency list (size 10+18 = 28 integers):
        0: 1, 2, 3, 4, 5, 6, 7, 8, 9,
        1: 0,
        2: 0,
        3: 0,
        4: 0,
        5: 0,
        6: 0,
        7: 0,
        8: 0,
        9: 0,
```



```
Graph with 10 vertices and 18 edges.
Degree sequence: [9, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency matrix (size 10x10 = 100 integers):
  0: 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
  1: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  2: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  3: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  4: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  5: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  6: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  7: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  8: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  9: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Star graph - 0 is connected out to all nodes except itself (starting at 0)

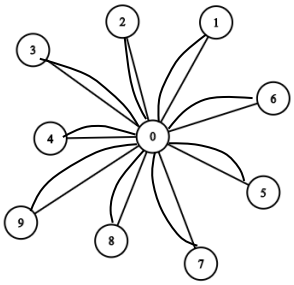
```
Graph with 10 vertices and 9 edges.
Degree sequence: [9, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency list (size 10+9 = 19 integers):
        0: 1, 2, 3, 4, 5, 6, 7, 8, 9,
        1:
        2:
        3:
        4:
        5:
        6:
        7:
        8:
        9:
```




```
Graph with 10 vertices and 9 edges.
Degree sequence: [9, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency matrix (size 10x10 = 100 integers):
  0: 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
  1: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  2: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  3: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  4: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  5: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  6: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  7: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  8: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  9: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Star graph - Each 'arm' consists of two undirected edges leading away from 0 and connecting to one vertex (starts at 0)

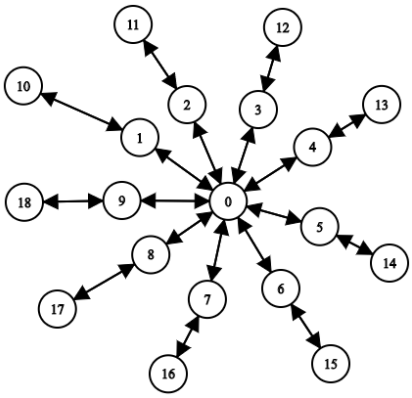
```
Graph with 10 vertices and 36 edges.
Degree sequence: [9, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency list (size 10+36 = 46 integers):
0: 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9,
1: 0, 0,
2: 0, 0,
3: 0, 0,
4: 0, 0,
5: 0, 0,
6: 0, 0,
7: 0, 0,
8: 0, 0,
9: 0, 0,
```




```
Graph with 10 vertices and 36 edges.
Degree sequence: [9, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency matrix (size 10x10 = 100 integers):
  0: 0, 2, 2, 2, 2, 2, 2, 2, 2, 2
  1: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  2: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  3: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  4: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  5: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  6: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  7: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  8: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
  9: 2, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Star graph - Each 'arm' is of length two and consists of undirected edges leading away from 0 (starts at 0)

```
Graph with 19 vertices and 36 edges.
Degree sequence: [9, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1].
Adjacency list (size 19+36 = 55 integers):
        0: 1, 2, 3, 4, 5, 6, 7, 8, 9,
        1: 10, 0,
        2: 11, 0,
        3: 12, 0,
        4: 13, 0,
        5: 14, 0,
        6: 15, 0,
        7: 16, 0,
        8: 17, 0,
        9: 18, 0,
        10: 1,
        11: 2,
        12: 3,
        13: 4,
        14: 5,
        15: 6,
        16: 7,
        17: 8,
        18: 9,
```




```
Graph with 19 vertices and 36 edges.
Degree sequence: [9, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Adjacency matrix (size 20x20 = 400 integers):
  0: 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  1: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
  2: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0
  3: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0
  4: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0
  5: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0
  6: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0
  7: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0
  8: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0
  9: 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0
 10: 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 11: 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 12: 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 13: 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 14: 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 15: 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 16: 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 17: 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 18: 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 19: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```