Lecture-9-Graph Distances-Intro

July 12, 2021

1 Lecture 9: Graph Distances

Today we will discuss Graph Distances and its applications particularly to the small world problem.

Suppose G is connected. If G is not connected, then apply the discussion below to a connected component of G.

- Given u, v two vertices in G, the **distance** $d(u, v)$ is the number of edges in the shortest path between u and v.
  - Observe that $d(u, v)$ is infinite if u and v are not in the same connected component. Hence we apply our discussion to connected graphs.
  - Moroever, $d(u, v) = d(v, u)$ since a path from v to u is also a path from u to v.
- The **eccentricity** $\epsilon(v)$ of a vertex is

$$\epsilon(v) = \max_{u \in V} d(v, u).$$

  In other words it is the distance between v and the farthest vertex from v.

- The **radius** r of a graph is the minimum eccentricity of any vertex,

$$r(G) = \min_{v \in V} \epsilon(v) = \min_{v \in V} \max_{u \in V} d(v, u).$$

- The **diameter** d of a graph is the maximum eccentricity of any vertex,

$$d(G) = \max_{v \in V} \epsilon(v) = \max_{v \in V} \max_{u \in V} d(v, u).$$

- Therefore equivalently

$$d(G) = \max_{u,v \in V} d(v, u).$$

  In other words, note down all the distances among all vertex combinations, then the diameter is the maximum among these.

- A **central** vertex is the vertex with

$$\epsilon(v) = r(G),$$

  the vertex whose maximum distance is the minimum among all vertices. (This coincides with our intuition of center).

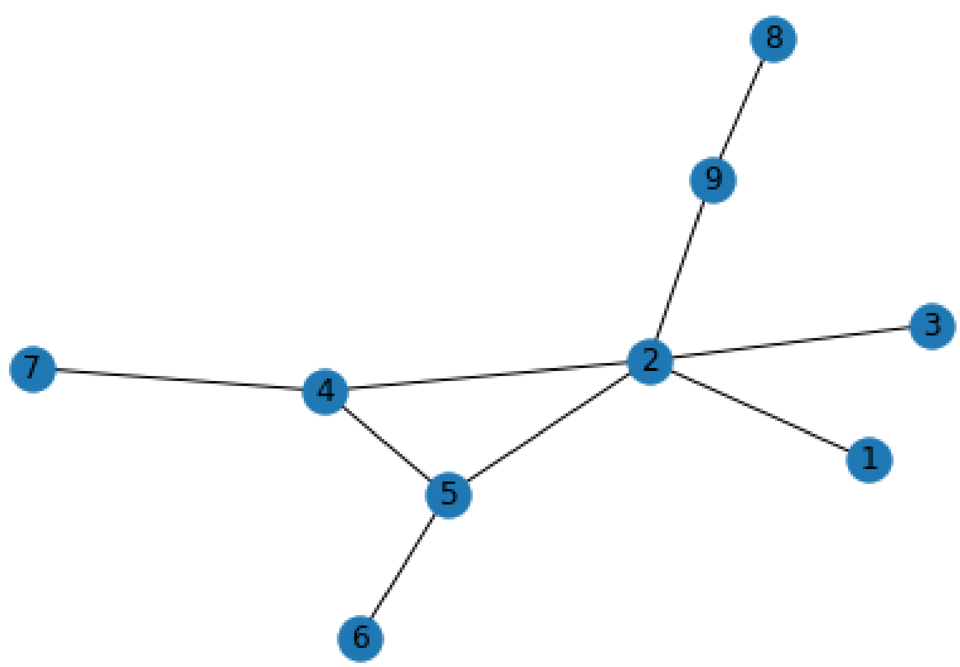
- A **peripheral** vertex is the vertex with

$$\epsilon(v) = d(G),$$

the vertex whose maximum distance is the maximum among all vertices.

[182]:
```
import networkx as nx
G = nx.Graph()

G.add_nodes_from([1,2,3,4,5,6,7,8,9])
G.add_edges_from([(1,2), (2,3), (2,4), (2,5), (4,5), (4,7), (8,9), (2,9),␣
↪(5,6)])
```

[183]:
```
nx.draw(G, with_labels=True)
```



Let us compute the distance between vertices 1 and 3

[184]:
```
nx.shortest_path_length(G,1,6)
```

[184]: 3

Let us find the eccentricity of a vertex say 1.

[185]:
```
nx.shortest_path_length(G,1)
```

[185]: {1: 0, 2: 1, 3: 2, 4: 2, 5: 2, 9: 2, 6: 3, 7: 3, 8: 3}

Outputs the distances from vertex 1 to other vertices in the graph as a dictionary. We need to find the maximum to get eccentricity.

[186]:
```
max(nx.shortest_path_length(G,1).values())
```

[186]: 3

Let us print eccentricities of all vertices

[187]:
```
ecc = {}
for v in G.nodes:
    ecc[v] = max(nx.shortest_path_length(G,v).values())
```

[188]:
```
ecc
```

[188]: {1: 3, 2: 2, 3: 3, 4: 3, 5: 3, 6: 4, 7: 4, 8: 4, 9: 3}

Therefore to find the radius of the graph, we need to find minimum in ecc.

[189]:
```
radiusG = min(ecc.values())
```

[190]:
```
radiusG
```

[190]: 2

Similarly to find diameter we need to find maximum in ecc

[191]:
```
diameterG = max(ecc.values())
```

[192]:
```
diameterG
```

[192]: 4

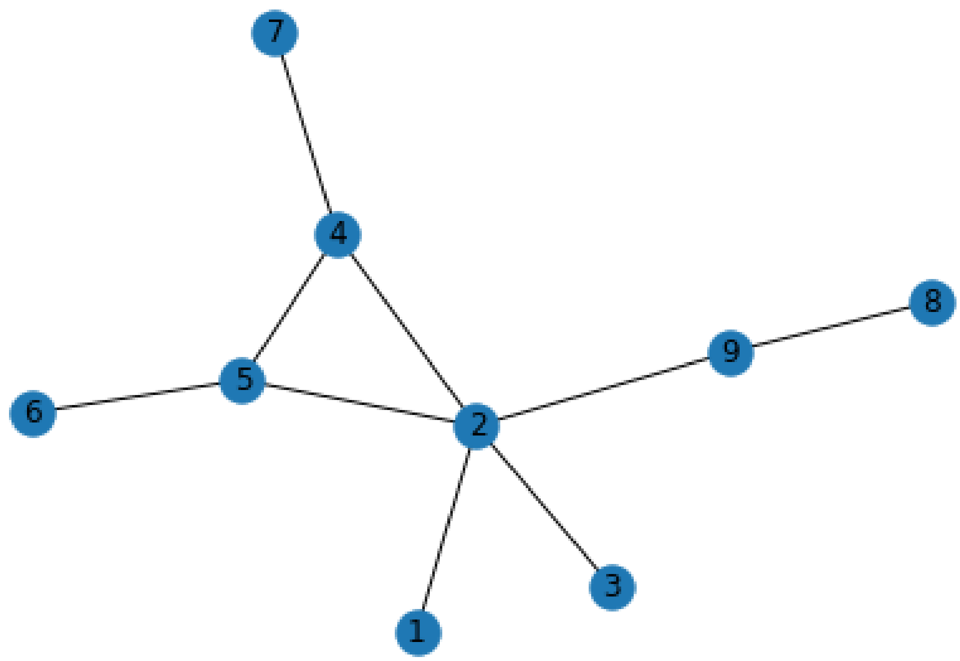
Find the central vertex

[193]:
```
ecc_values = list(ecc.values())
ecc_nodes = list(ecc.keys())

centerG = ecc_nodes[ecc_values.index(min(ecc_values))]
periG = ecc_nodes[ecc_values.index(max(ecc_values))]

print(centerG)
print(periG)
```

```
2
6
```

As evident vertex 2 is the central vertex, and 6 is a peripheral vertex.

```
[194]: nx.draw(G, with_labels=True)
```



Most of these can be done with `networkx` as well, for example

```
[195]: nx.eccentricity(G)
```

```
[195]: {1: 3, 2: 2, 3: 3, 4: 3, 5: 3, 6: 4, 7: 4, 8: 4, 9: 3}
```

```
[196]: nx.diameter(G)
```

```
[196]: 4
```

```
[197]: nx.radius(G)
```

```
[197]: 2
```

```
[198]: nx.periphery(G)
```

```
[198]: [6, 7, 8]
```

```
[199]: nx.center(G)
```

[199]: [2]

1.0.1 Average Distance

Suppose we ask another question. If I pick two vertices uniformly from the G what is the expected distance?

In other words, on average what is the distance between two vertices in the graph. We can compute it as

[111]:
```
def meanDistance(G):
    import statistics
    distanceG = [nx.shortest_path_length(G, u, v) for u in G.nodes for v in G.
 ↪nodes]
    return statistics.mean(distanceG)

def medianDistance(G):
    import statistics
    distanceG = [nx.shortest_path_length(G, u, v) for u in G.nodes for v in G.
 ↪nodes]
    return statistics.median(distanceG)
```

[112]:
```
meanDistance(G)
```

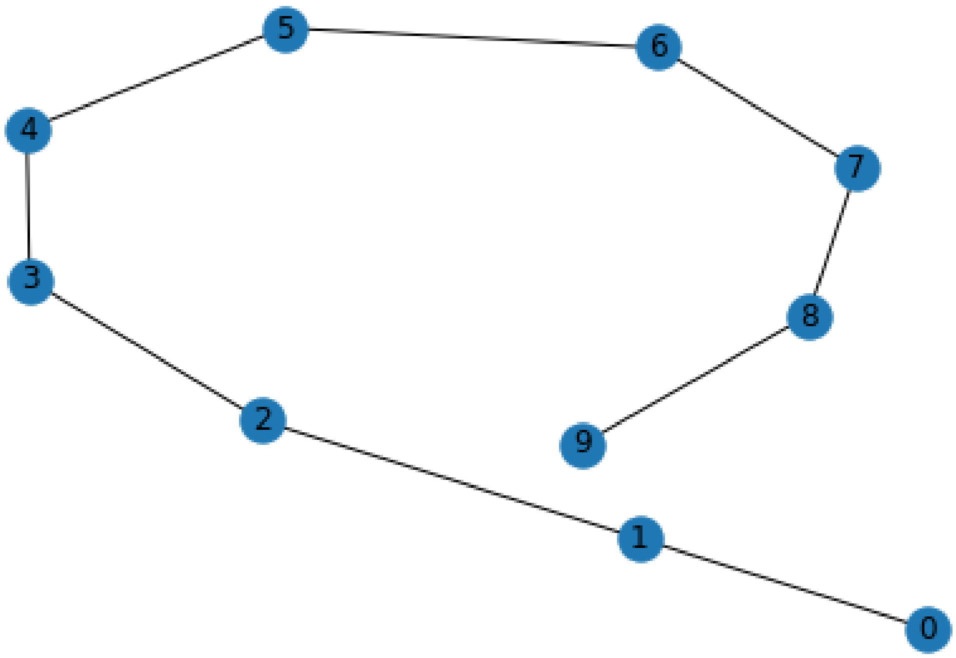
[112]: 1.9259259259259258

[114]:
```
def detailsGraph(H):
    print("Radius", nx.radius(H))
    print("Diameter", nx.diameter(H))
    print("Peripheral Vertices", nx.periphery(H))
    print("Central Vertices", nx.center(H))
    print("Mean Distance", meanDistance(H))
    print("Median Distance", medianDistance(H))
```

1.1 Examples of Graphs and their distances

[138]:
```
H = nx.path_graph(10)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

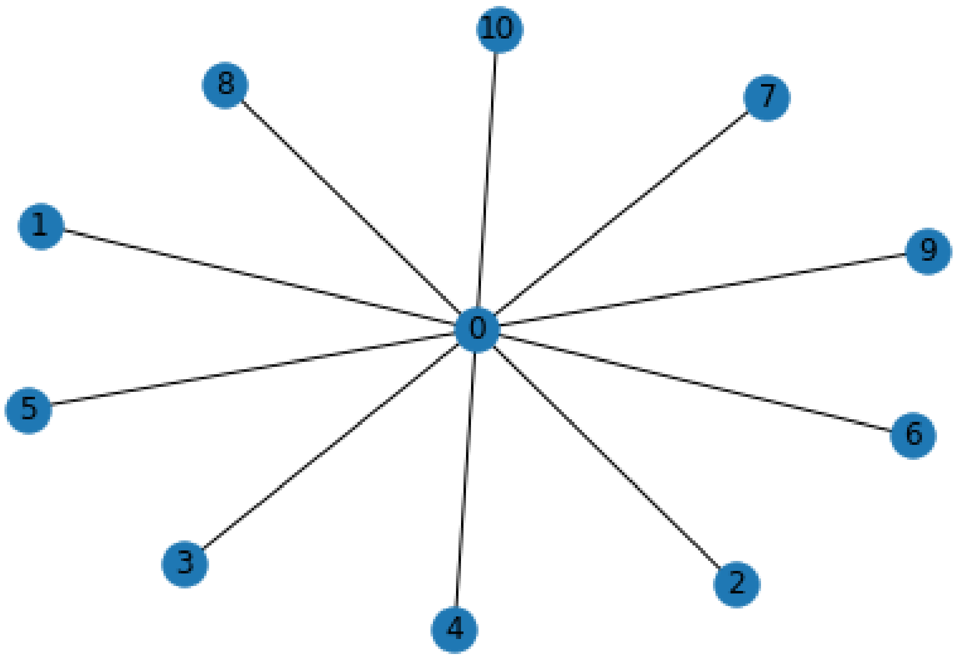
```
Radius 5
Diameter 9
Peripheral Vertices [0, 9]
Central Vertices [4, 5]
Mean Distance 3.3
Median Distance 3.0
```



[139]:

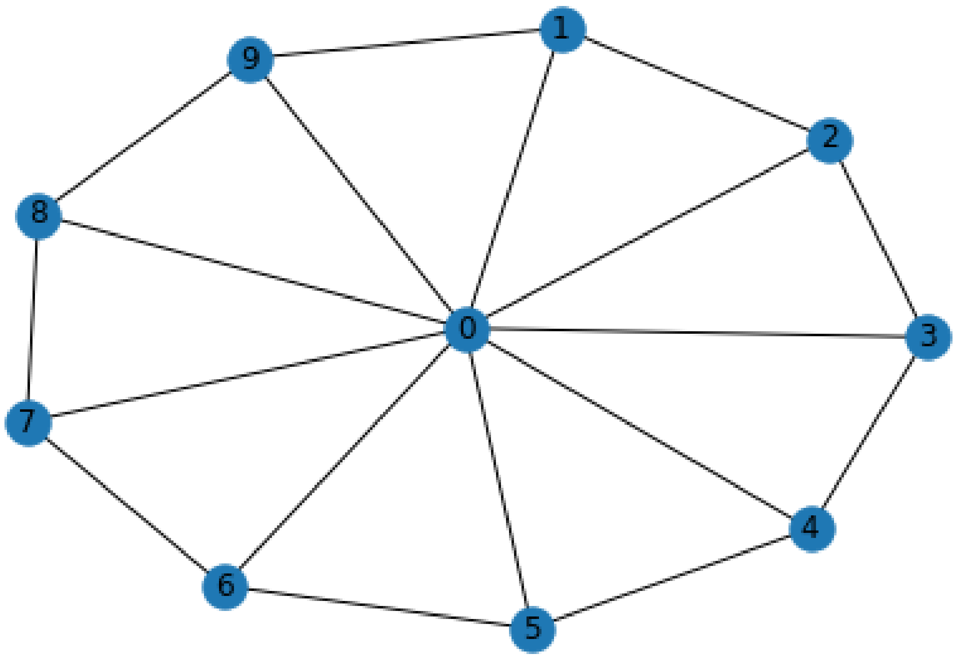
```
H = nx.star_graph(10)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

```
Radius 1
Diameter 2
Peripheral Vertices [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Central Vertices [0]
Mean Distance 1.6528925619834711
Median Distance 2
```



[140]:
```
H = nx.wheel_graph(10)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

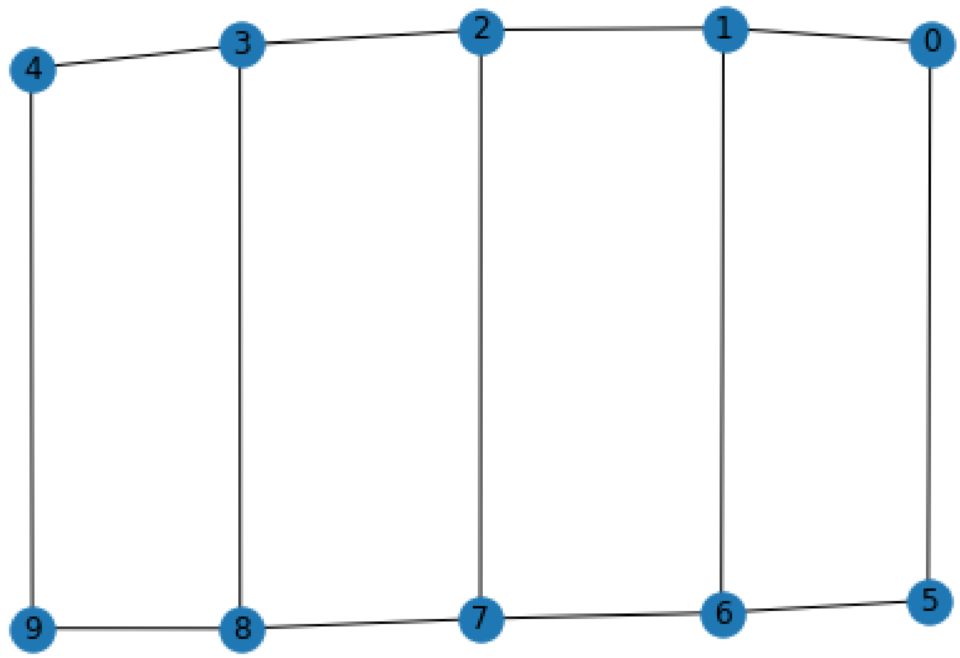
```
Radius 1
Diameter 2
Peripheral Vertices [1, 2, 3, 4, 5, 6, 7, 8, 9]
Central Vertices [0]
Mean Distance 1.44
Median Distance 2.0
```



We can do our analysis on predetermined graphs in networkx. As a digression, let us look at a few predetermined graphs in networkx

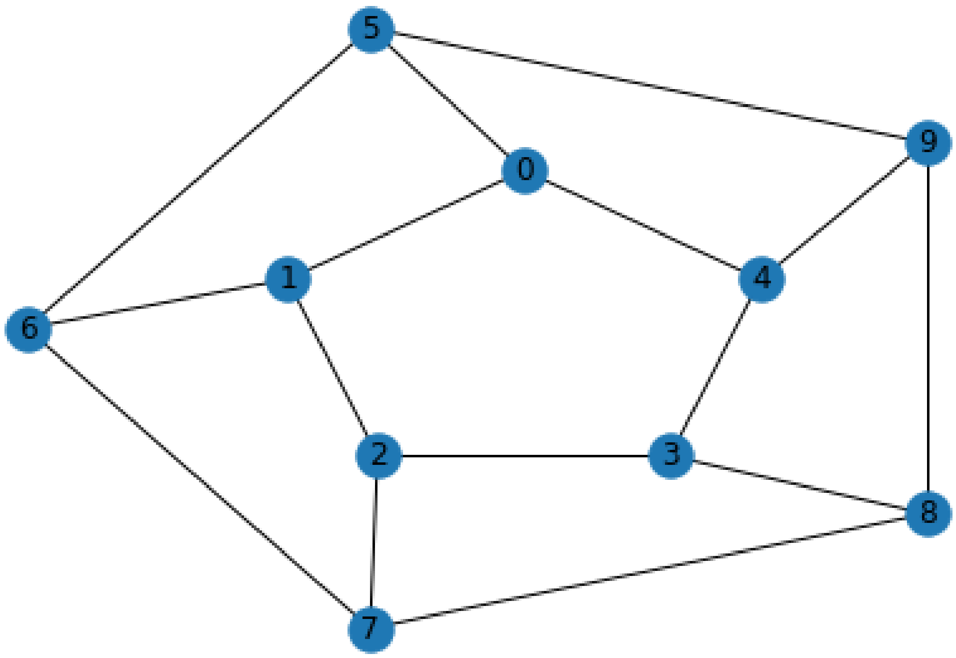
[141]:
```
H = nx.ladder_graph(5)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

```
Radius 3
Diameter 5
Peripheral Vertices [0, 4, 5, 9]
Central Vertices [2, 7]
Mean Distance 2.1
Median Distance 2.0
```



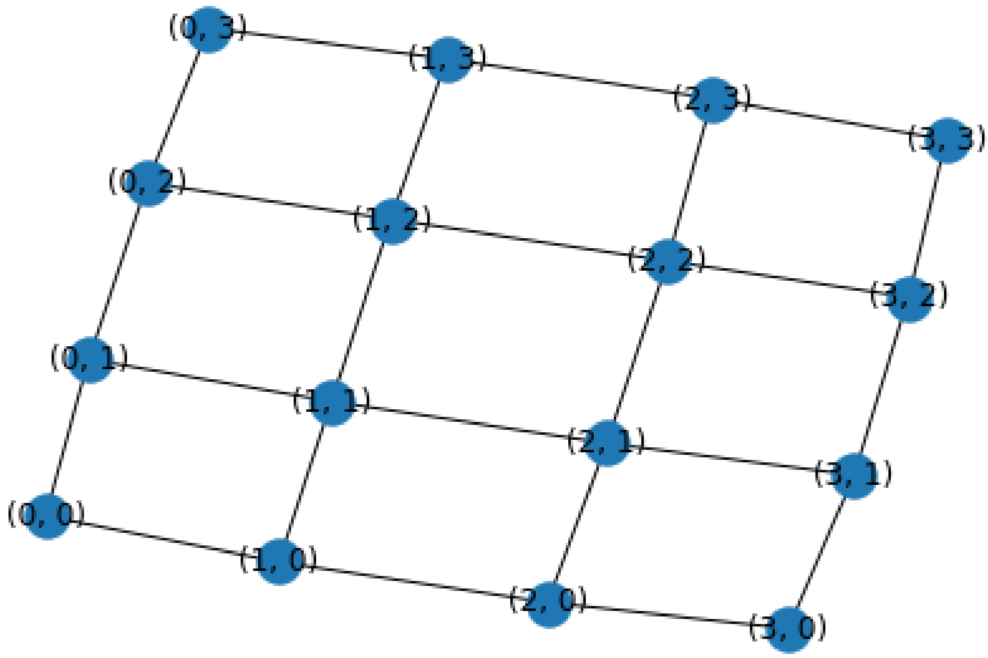
[142]:
```
H = nx.circular_ladder_graph(5)
pos = nx.shell_layout(H, nlist=[[0,1,2,3,4],[6,7,8,9,5]])
nx.draw(H,pos, with_labels=True)
detailsGraph(H)
```

```
Radius 3
Diameter 3
Peripheral Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Central Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Mean Distance 1.7
Median Distance 2.0
```



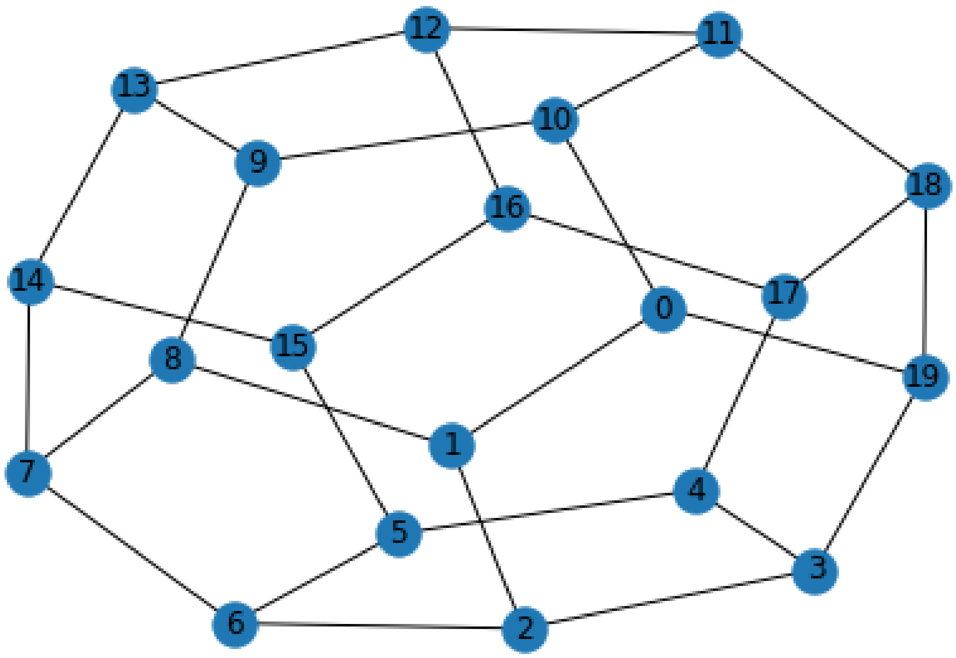
[166]:
```
H = nx.grid_2d_graph(4,4)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

```
Radius 4
Diameter 6
Peripheral Vertices [(0, 0), (0, 3), (3, 0), (3, 3)]
Central Vertices [(1, 1), (1, 2), (2, 1), (2, 2)]
Mean Distance 2.5
Median Distance 2.0
```



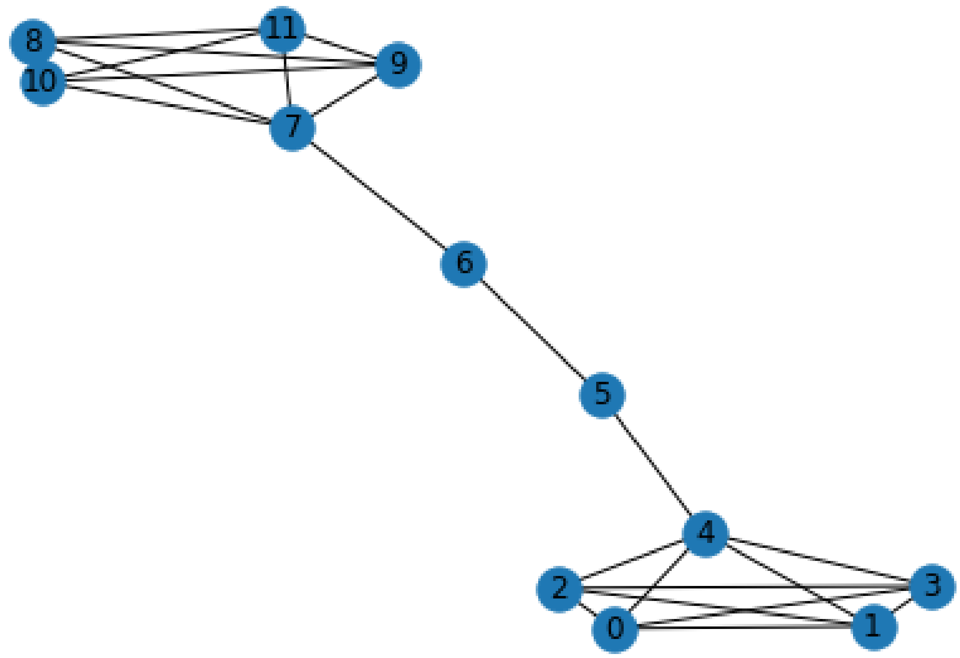
[168]:
```
H = nx.dodecahedral_graph()
nx.draw(H, with_labels=True)
detailsGraph(H)
```

```
Radius 5
Diameter 5
Peripheral Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19]
Central Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19]
Mean Distance 2.5
Median Distance 2.5
```



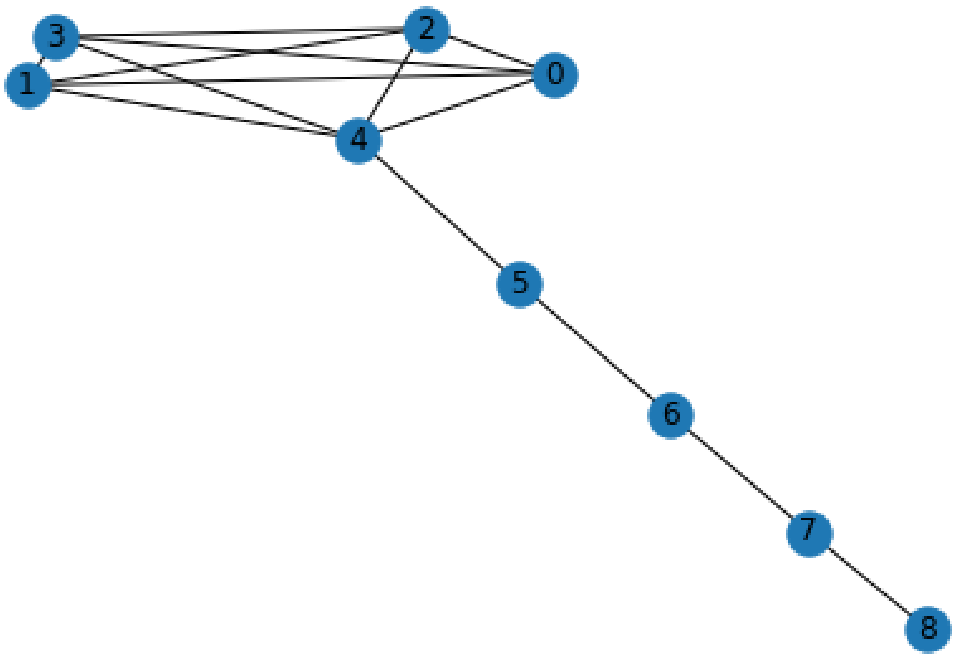
[143]:
```
H = nx.barbell_graph(5,2)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

```
Radius 3
Diameter 5
Peripheral Vertices [0, 1, 2, 3, 8, 9, 10, 11]
Central Vertices [5, 6]
Mean Distance 2.5277777777777777
Median Distance 2.0
```



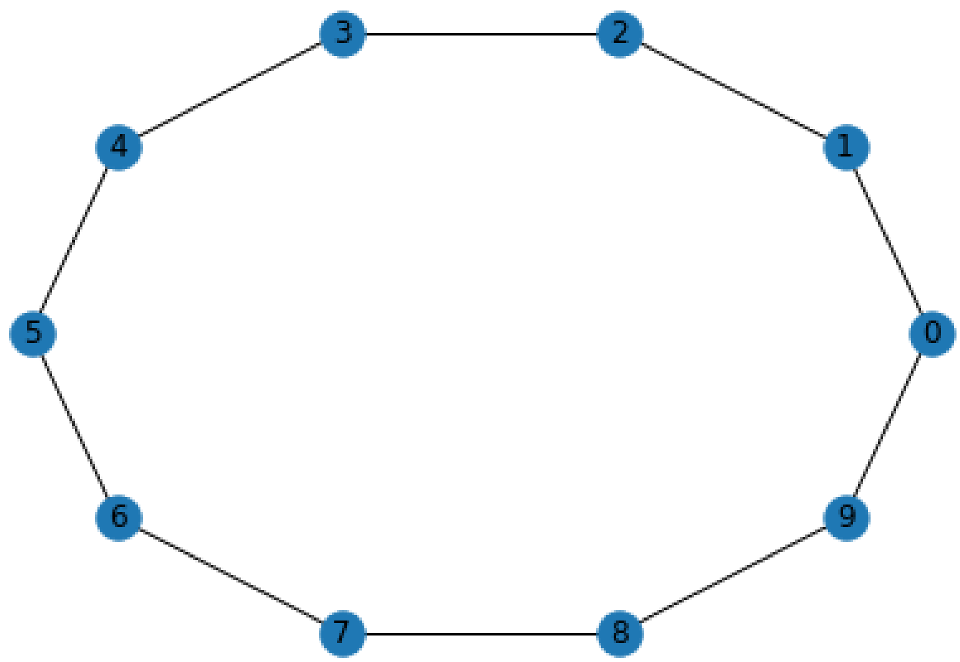
[144]:
```
H = nx.lollipop_graph(5,4)
nx.draw(H, with_labels=True)
detailsGraph(H)
```

```
Radius 3
Diameter 5
Peripheral Vertices [0, 1, 2, 3, 8]
Central Vertices [5, 6]
Mean Distance 2.123456790123457
Median Distance 2
```



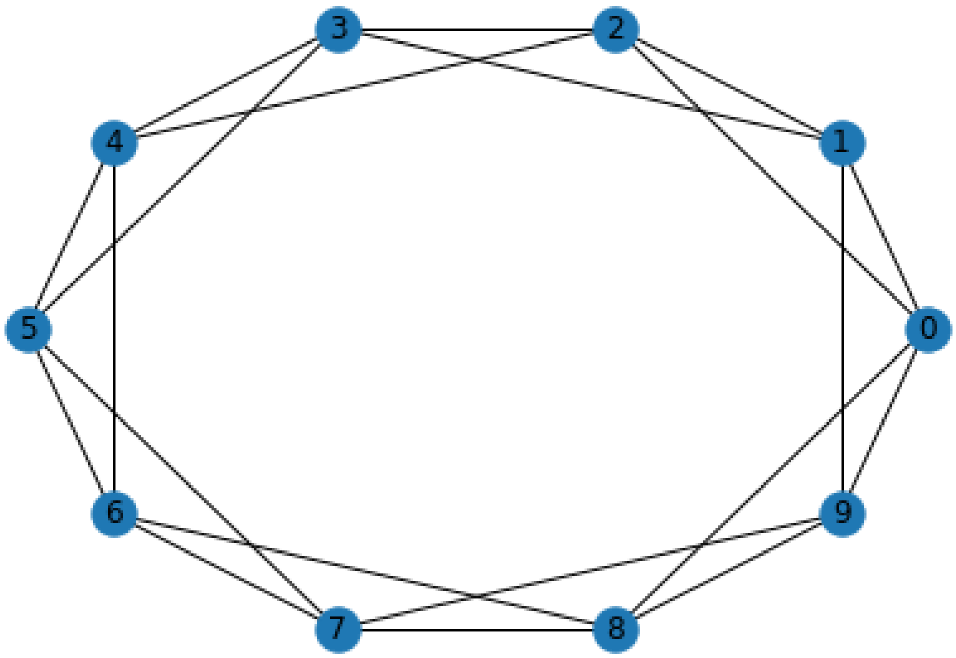
[145]:
```
H = nx.circulant_graph(10,[1])
pos = nx.circular_layout(H)
nx.draw(H,pos, with_labels=True)
detailsGraph(H)
```

```
Radius 5
Diameter 5
Peripheral Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Central Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Mean Distance 2.5
Median Distance 2.5
```



[146]:
```
H = nx.circulant_graph(10,[1,2])
pos = nx.circular_layout(H)
nx.draw(H,pos, with_labels=True)
detailsGraph(H)
```

```
Radius 3
Diameter 3
Peripheral Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Central Vertices [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Mean Distance 1.5
Median Distance 1.5
```



1.2 Application

The Small World Problem.

In 1929, Frigyes Karinthy (a Hungarian writer) proposed the six degrees of freedom concept, in his short story Chains

He observes that the population of earth is more connected than ever before. He proclaimed that any two individuals can be connected to one another through at most **five** acquaintances. He suggested performing the following experiment:

> A fascinating game grew out of this discussion. One of ussuggested performingthe following experiment toprove that the population of the Earth is closer together now than they haveever been before. We should select any person from the 1.5 billion inhabitants of the Earth – anyone, anywhere at all. He bet us that, using no more than five individuals, one of whom is a personal acquaintance, he could contact the selected individual using nothing except the network of personal acquaintances.

He asked another question: >Was there ever a time inhuman historywhen this would have been impossible?

> Julius Caesar, for instance, was a popular man, but if he had got it intohis head to try and contact a priest from one of the Mayan orAztec tribes thatlived in the Americas at that time, he could nothave succeeded -not in five steps, not even in three hundred.

1.3 Mathematical Interpretation

Note that if we draw a graph where vertices stand for people, and we connect vertex u to vertex v if u and v know each other (Here we assume that if u and v know each other, then so do v and u as well, in other words acquaintance is symmetric).

Then what Frigyes was saying that the diameter of this graph is at most 6.

1.4 Experimental Study

Stanley Milgram and other researchers performed numerous experiments in the 1960s to confirm/deny Frigues' Question.

Jeffrey Travers and Stanley Milgram published a seminal paper An Experimental Study of Small World Problem in 1969.

For more lets read here

In brief they found that the average distance was 5.2

This seemed to confirm Frigues' Hypothesis.

1.5 Extensions of the Small-World Idea

- Suppose our graph consists of Chess Players instead of all people. We connect u and v by an edge if u and v have played a chess game.

  Instead of finding the mean distance or diameter, suppose we want to find distance between say Magnus Carlsen and any other Chess Player v, ie

  $$d(\text{Carlsen}, v).$$

  There is a notion of this known as Morphy Number where we measure the distance between chess players and the 19th century chess maestor Paul Morphy.

- Mathematicians use a similar number knows as Erdös number. Here an edge connects u and v if mathematican u has collaborated with v. We want to find

  $$d(\text{Erdös}, v).$$

  For example Einstein has an Erdös number of 2 via (A. Einstein – E. Straus – P. Erdös)

- Similarly, Hollywood uses Bacon Number.

[ ]: