- **How many items are present in the network and how many co-purchases happened?**

```
[4]: print(nx.info(G_amazon))

Name:
Type: DiGraph
Number of nodes: 2647
Number of edges: 10841
Average in degree:   4.0956
Average out degree:   4.0956
```

There are 2647 items in the network, and 10841 co-purchases happened.

- **Compute the average shortest distance between the nodes in graph G. Explain your results briey.**

The average shortest distance is 11.9934

Compute the average shortest distance

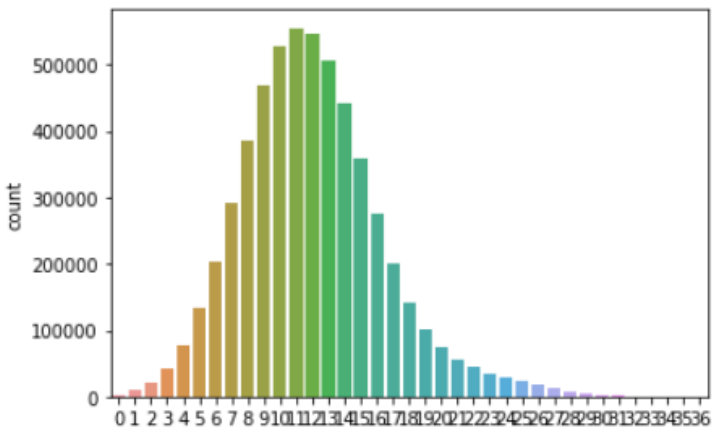
1. compute the shortest distance to other nodes for each node
2. compute the average

```
[5]: shortest_distance = []
for v in G_amazon.nodes():
    spl = nx.single_source_shortest_path_length(G_amazon, v)
    for p in spl.values():
        shortest_distance.append(p)
average_shortest_distance = sum(shortest_distance) / len(shortest_distance)
print('The average shortest distance is {:.4f}'.format(average_shortest_distance))

The average shortest distance is 11.9934

]: sns.countplot(shortest_distance)

]: <matplotlib.axes._subplots.AxesSubplot at 0x181f5c9b828>
```



Most of the nodes in the graph can reach any node in the whole graph through 12 nodes on average.

- **Compute the transitivity and the average clustering coefficient of the network graph G. Explain your findings briefly based on the definitions of clustering coefficient and transitivity.**

The transitivity of the network graph is 0.4339.

The average clustering coefficient of the network graph is 0.4086

```
transitivity = nx.transitivity(G_amazon)
print('The transitivity of the network graph is {:.4f}'.format(transitivity))
average_clustering_coefficient = nx.average_clustering(G_amazon)
print('The average clustering coefficient of the network graph is {:.4f}'.format(average_clustering_coefficient))
```

```
The transitivity of the network graph is 0.4339
The average clustering coefficient of the network graph is 0.4086
```

The values of transitivity and average clustering coefficient about this network graph are a little lower which means the connection of this graph is not so dense, the graph is not particularly complicated.

- **Apply the PageRank algorithm to network G with damping value 0.5 and find the 10 nodes with the highest PageRank. Explain your findings briey.**

  The top 10 item with the highest pagerank is [8, 481, 33, 18, 23, 30, 346, 99, 93, 21]

Compute the pagerank and find 10 nodes with the highest pagerank

```
pagerank = nx.pagerank(G_amazon, alpha=0.5)
pagerank_sorted = sorted(pagerank.items(), key=lambda x: x[1], reverse=True)
highest_10 =  [i[0] for i in pagerank_sorted[:10]]
print('The top 10 item with the highest pagerank is {}'.format(highest_10))
```

```
The top 10 item with the highest pagerank is [8, 481, 33, 18, 23, 30, 346, 99, 93, 21]
```

The larger the value of pagerank, the more connections the node has with

other nodes in the graph, and the greater the importance of the node in the graph. So these ten nodes are especially important for this graph, and they are mostly co-purchased with other items. These may be an indispensable commodity in life.