

In [1]:

```
import numpy as np # to handle numeric data
import matplotlib.pyplot as plt # for visualization
import pandas as pd # for handling dataframe
```

In [2]:

```
ourData = pd.read_csv('Mall_Customers.csv') # read the data
ourData.head() # print the first five rows of our dataset
```

Out[2]:

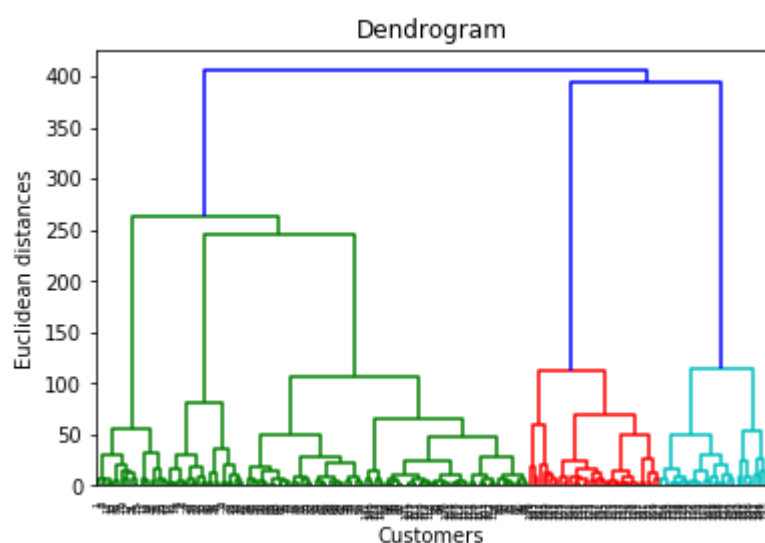
	CustomerID	Genre	Age	Annual Income (k\$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [3]:

```
newData = ourData.iloc[:, [3, 4]].values # extract the two features from our dataset
```

In [5]:

```
import scipy.cluster.hierarchy as sch # importing scipy.cluster.hierarchy for dendrogram
dendrogram = sch.dendrogram(sch.linkage(newData, method = 'ward')) # finding the optimal nu
plt.title('Dendrogram') # title of the dendrogram
plt.xlabel('Customers') # label of the x-axis
plt.ylabel('Euclidean distances') # label of the y-axis
plt.show() # show the dendrogram
```



In [7]:

```
from sklearn.cluster import AgglomerativeClustering # this line of code imports AgglomerativeClustering
'''
we need to create an AgglomerativeClustering object, and in it, we pass the following parameters:
n_clusters= 5, the number of clusters our model should return
affinity=euclidean, specify metric to be used to calculate distances
linkage= ward to regulate how distance calculation will be carried out between different clusters
'''
Agg_hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
y_hc = Agg_hc.fit_predict(newData) # model fitting on the dataset
```

In [8]:

```
# plotting cluster 1
plt.scatter(newData[y_hc == 0, 0], newData[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(newData[y_hc == 1, 0], newData[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(newData[y_hc == 2, 0], newData[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(newData[y_hc == 3, 0], newData[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(newData[y_hc == 4, 0], newData[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
# plot title addition
plt.title('Clusters of customers')
# Labelling the x-axis
plt.xlabel('Annual Income (k$)')
# Label of the y-axis
plt.ylabel('Spending Score (1-100)')
# printing the legend
plt.legend()
# show the plot
plt.show()
```



In [ ]: