In [1]:
```
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

iris = load_iris()
X = iris.data

kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
kmeans.labels_
```

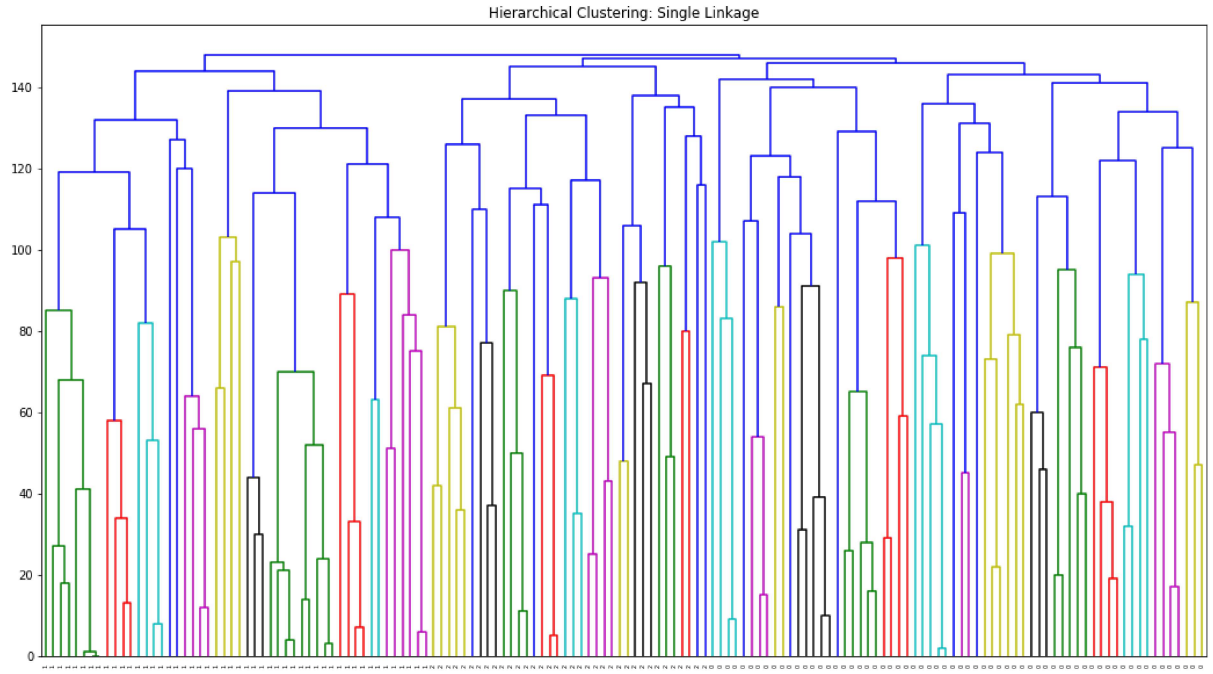
Out[1]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
       2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

In [2]:
```
from matplotlib import pyplot as plt
from scipy.cluster.hierarchy import dendrogram
import numpy as np

def plot_dendrogram(model, **kwargs):
    children = model.children_
    distance = np.arange(children.shape[0])
    no_of_observations = np.arange(2, children.shape[0]+2)
    linkage_matrix = np.column_stack([children, distance, no_of_observations]).ast
    dendrogram(linkage_matrix, **kwargs)
```

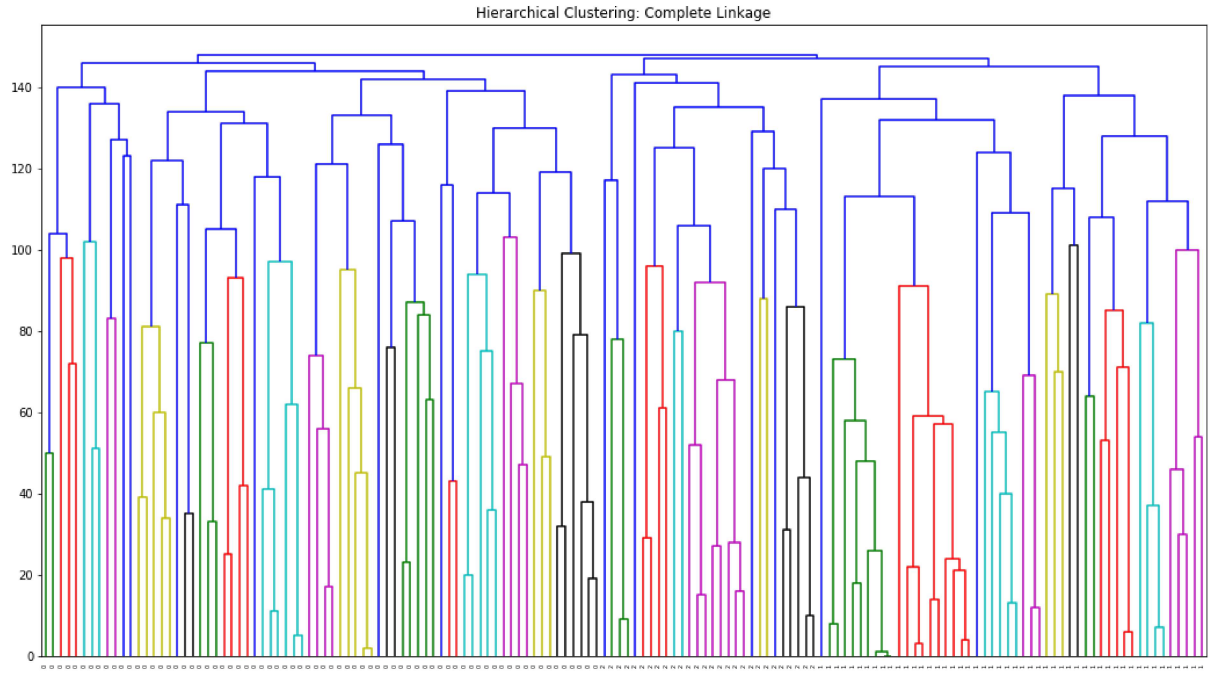
In [3]:
```
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(linkage="ward",n_clusters=3)
singleLinkage = model.fit(X)

plt.figure(figsize=(18,10))
plt.title('Hierarchical Clustering: Single Linkage')
plot_dendrogram(model, labels=singleLinkage.labels_)
plt.show()
```



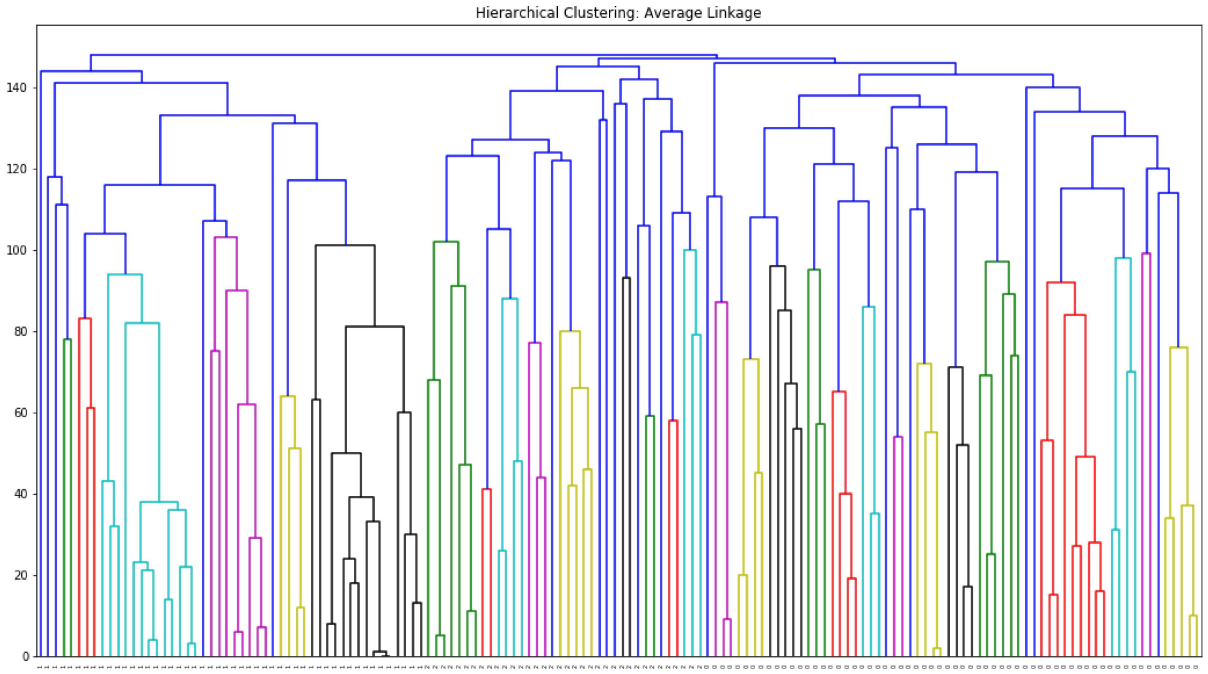
In [4]:
```
model = AgglomerativeClustering(linkage="complete",n_clusters=3)
completeLinkage = model.fit(X)

plt.figure(figsize=(18,10))
plt.title('Hierarchical Clustering: Complete Linkage')
plot_dendrogram(model, labels=completeLinkage.labels_)
plt.show()
```



In [5]:
```
model = AgglomerativeClustering(linkage="average",n_clusters=3)
averageLinkage = model.fit(X)

plt.figure(figsize=(18,10))
plt.title('Hierarchical Clustering: Average Linkage')
plot_dendrogram(model, labels=averageLinkage.labels_)
plt.show()
```



In [ ]:

In [ ]: