Applied Hierarchical Clustering (Agglomerative) to the given datapoints

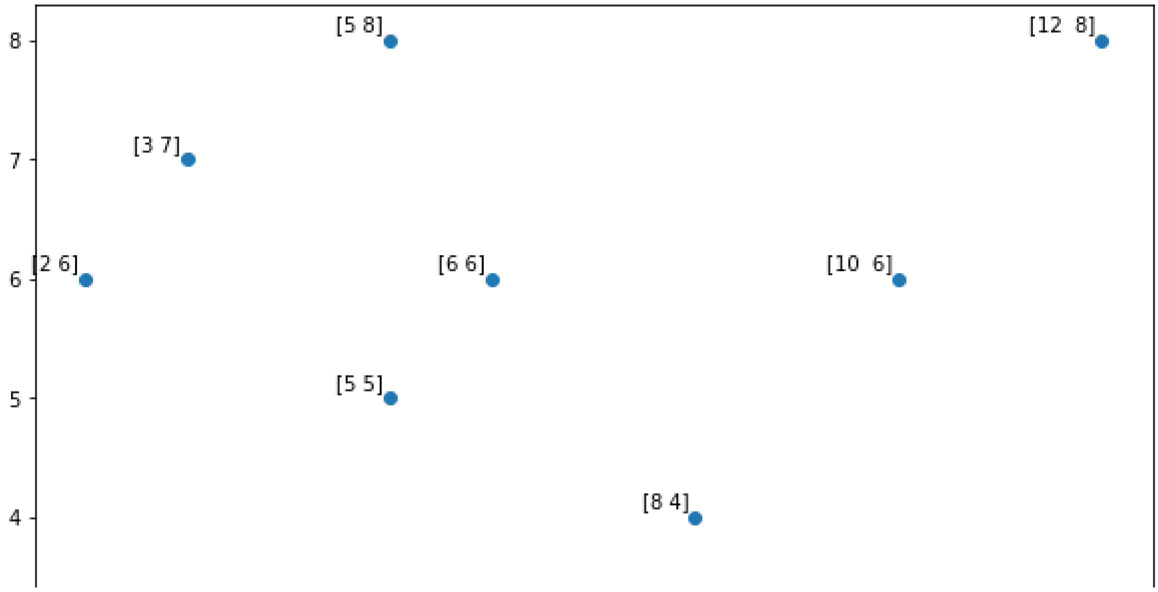
Also, plotted single link, average link, and complete link clusters of the given datapoints.

```
#importing libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
```

```
#created an array of given datapoints
X = np.array([[3, 7],
[2, 6],
[2, 2],
[5, 2],
[7, 3],
[8, 4],
[10, 6],
[12, 8],
[5, 5],
[6, 6],
[5, 8]])
```

```
#plotting datapoints
labels = X

plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
    label,
    xy=(x, y), xytext=(-3, 3),
    textcoords='offset points', ha='right', va='bottom')
plt.show()
```

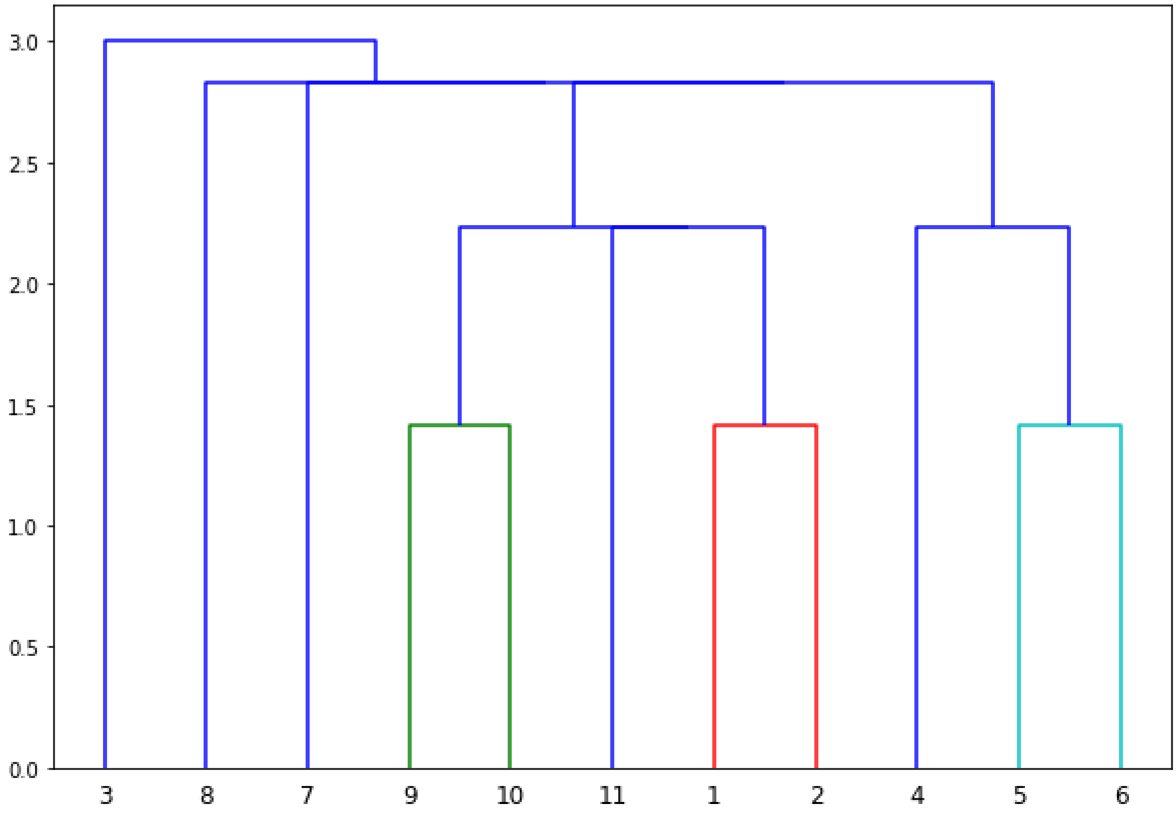
```
/usr/local/lib/python3.7/dist-packages/matplotlib/text.py:1165: FutureWarning: elemen
  if s != self._text:
```



```
#plotting datapoints with labels from 1 to 12
labels = range(1, 12)

plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
    label,
    xy=(x, y), xytext=(-3, 3),
    textcoords='offset points', ha='right', va='bottom')
plt.show()
```

```
#plotting single link cluster
print ("\n single link cluster \n")

linked = linkage(X, 'single')
labelList = range(1, 12)
#labelList = X
plt.figure(figsize=(10, 7))

dendrogram(linked,
orientation='top',
labels=labelList,
distance_sort='descending',
show_leaf_counts=True)

plt.show()
```

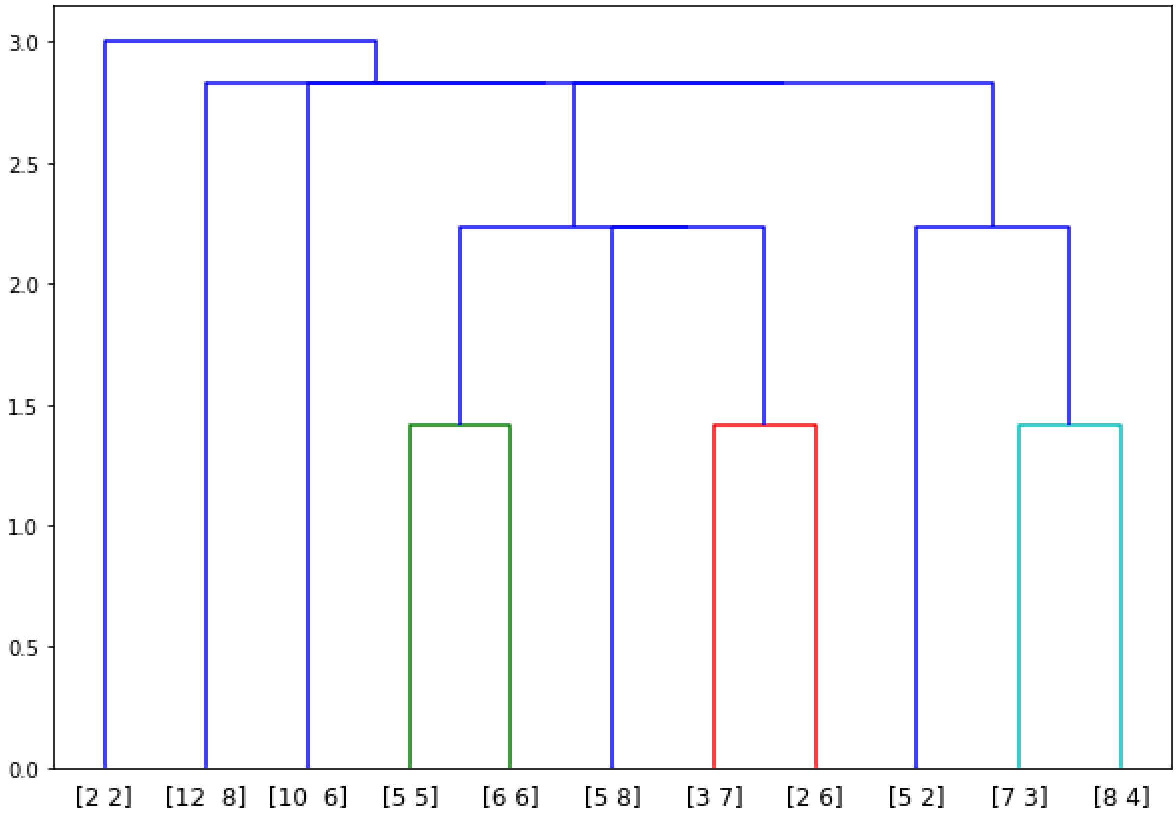
```
 single link cluster 
```



```
#plotting single link cluster
print ("\n single link cluster \n")

linked = linkage(X, 'single')
#labelList = range(1, 12)
labelList = X
plt.figure(figsize=(10, 7))
```

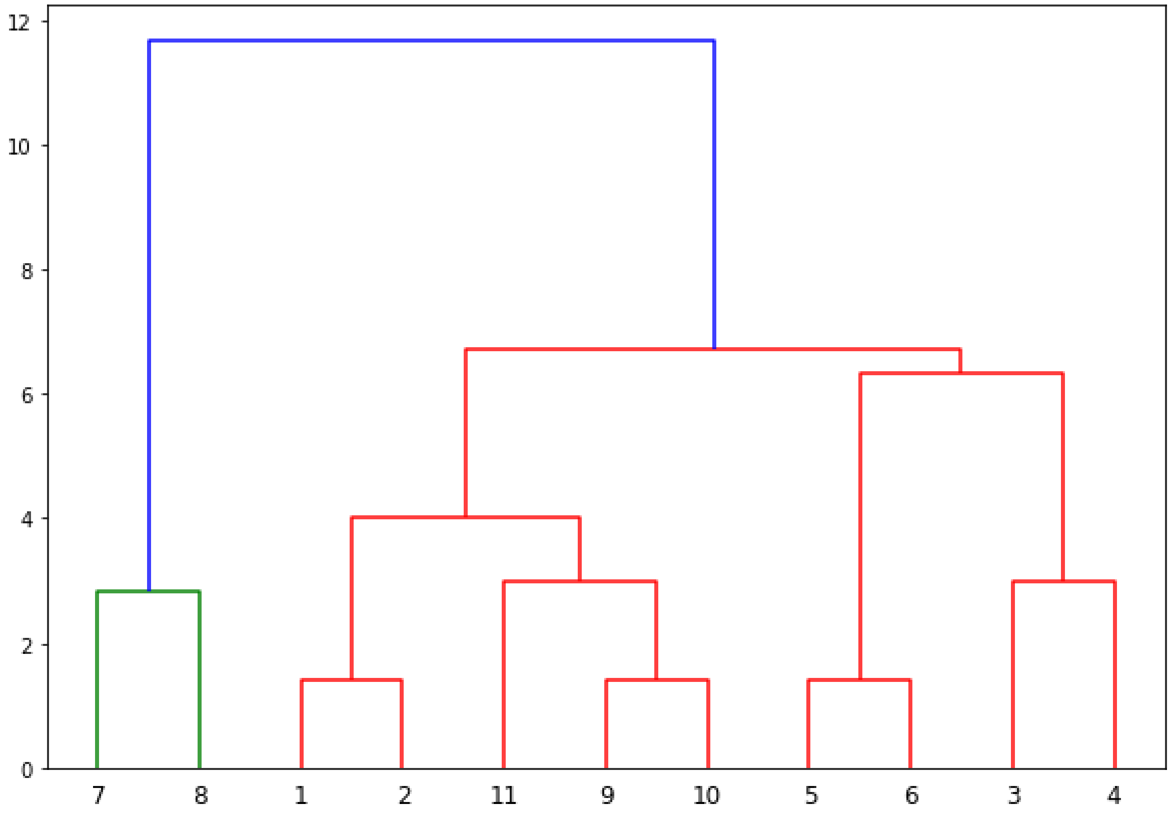
```
dendrogram(linked,
orientation='top',
labels=labelList,
distance_sort='descending',
show_leaf_counts=True)

plt.show()
```

```
  single link cluster

/usr/local/lib/python3.7/dist-packages/matplotlib/text.py:1165: FutureWarning: elemer
  if s != self._text:
```



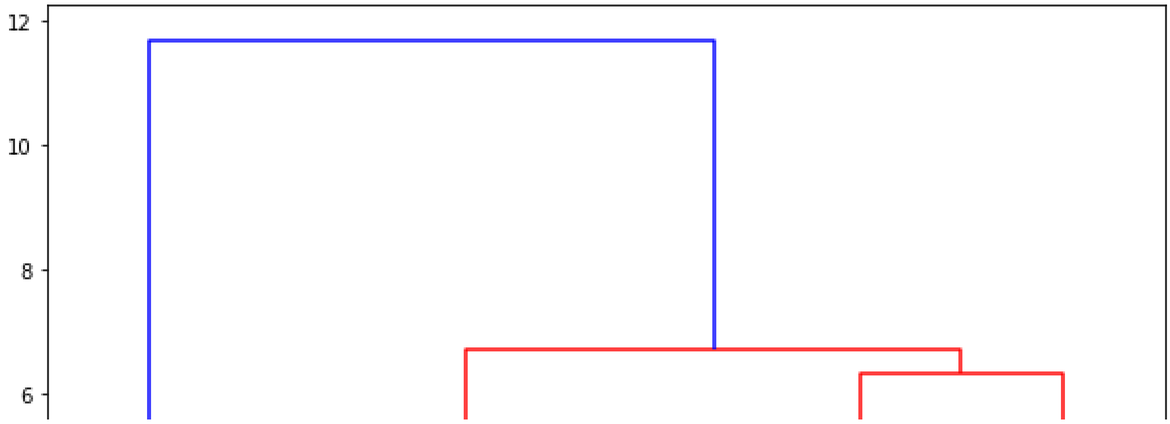
```
#plotting complete link cluster
print ("\n complete link cluster \n")

linked_complete = linkage(X, 'complete')
labelList = range(1, 12)
#labelList = X
plt.figure(figsize=(10, 7))

dendrogram(linked_complete,
orientation='top',
labels=labelList,
distance_sort='descending',
show_leaf_counts=True)

plt.show()
```

```
complete link cluster
```



```
#plotting complete link cluster
print ("\n complete link cluster \n")

linked_complete = linkage(X, 'complete')
#labelList = range(1, 12)
labelList = X
plt.figure(figsize=(10, 7))

dendrogram(linked_complete,
orientation='top',
labels=labelList,
distance_sort='descending',
show_leaf_counts=True)

plt.show()
```

```
 complete link cluster

/usr/local/lib/python3.7/dist-packages/matplotlib/text.py:1165: FutureWarning: elemen
  if s != self._text:
```

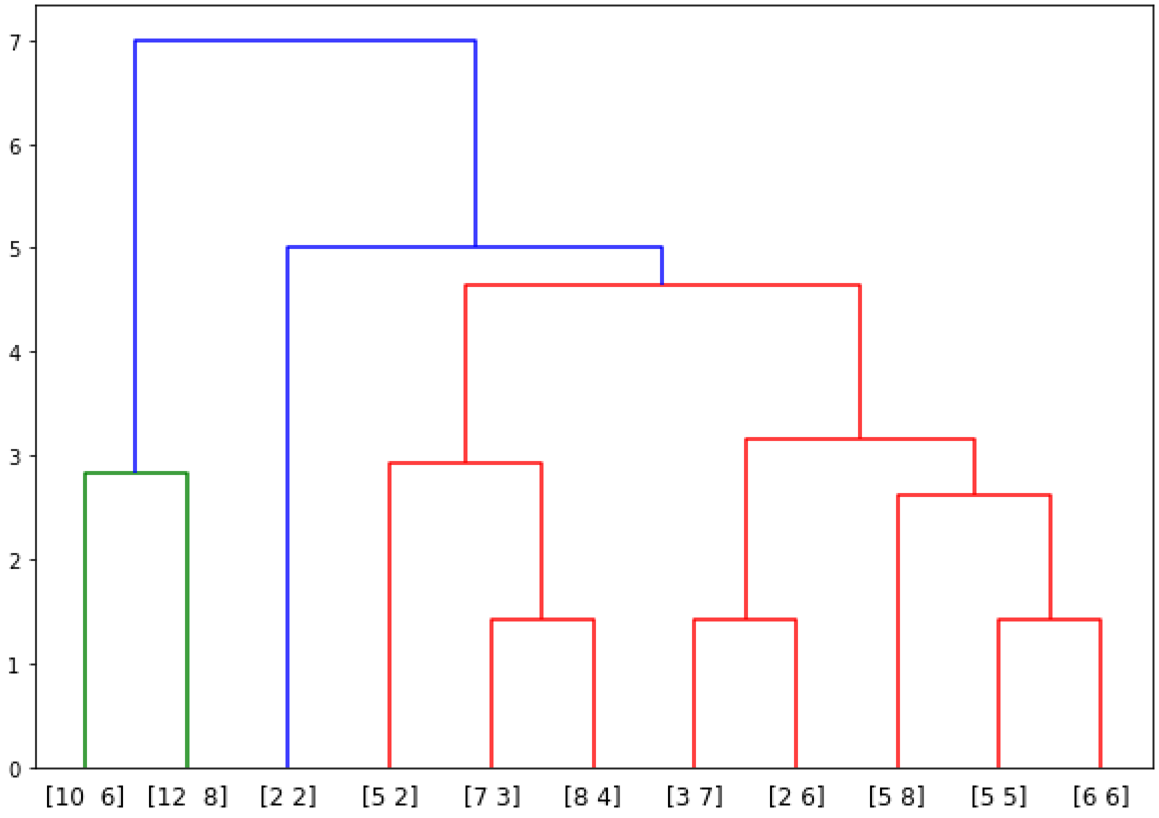


```
#plotting average link cluster
print ("\n average link cluster \n")

linked_average = linkage(X, 'average')
labelList = range(1, 12)
#labelList = X
plt.figure(figsize=(10, 7))

dendrogram(linked_average,
orientation='top',
labels=labelList,
distance_sort='descending',
show_leaf_counts=True)

plt.show()
```

```
      average link cluster
```

```
#plotting average link cluster
print ("\n average link cluster \n")

linked_average = linkage(X, 'average')
#labelList = range(1, 12)
labelList = X
plt.figure(figsize=(10, 7))

dendrogram(linked_average,
orientation='top',
labels=labelList,
distance_sort='descending',
show_leaf_counts=True)

plt.show()
```

```
      average link cluster

     /usr/local/lib/python3.7/dist-packages/matplotlib/text.py:1165: FutureWarning: elemen
       if s != self._text:
```



✓ 0s completed at 8:10 PM ● ×