1.Import Libraries

In [41]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

2.Import Dataset

In [92]:
```
data=pd.read_csv('Universities.csv')
data.head()
```

Out[92]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	Brown	1310	89	22	13	22704	94
1	CalTech	1415	100	25	6	63575	81
2	CMU	1260	62	59	9	25026	72
3	Columbia	1310	76	24	12	31510	88
4	Cornell	1280	83	33	13	21864	90

Data Undestanding

In [86]:
```
data.shape
```

Out[86]:
```
(25, 6)
```

In [88]:
```
data.isnull().sum()
```

Out[88]:
```
SAT         0
Top10       0
Accept      0
SFRatio     0
Expenses    0
GradRate    0
dtype: int64
```

In [90]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   SAT        25 non-null     float64
 1   Top10      25 non-null     float64
 2   Accept     25 non-null     float64
 3   SFRatio    25 non-null     float64
 4   Expenses   25 non-null     float64
 5   GradRate   25 non-null     float64
dtypes: float64(6)
memory usage: 1.3 KB
```

In [93]:
```
data=data.drop('Univ',axis=1)
data.head()
```

Out[93]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	1310	89	22	13	22704	94
1	1415	100	25	6	63575	81
2	1260	62	59	9	25026	72
3	1310	76	24	12	31510	88
4	1280	83	33	13	21864	90

4.Normlize Function

In [72]:
```
def norm_fun(i):
    x=(i-i.max())/(i.max()-i.min())
    return x
```

In [94]:
```
data=norm_fun(data)
data.head()
```

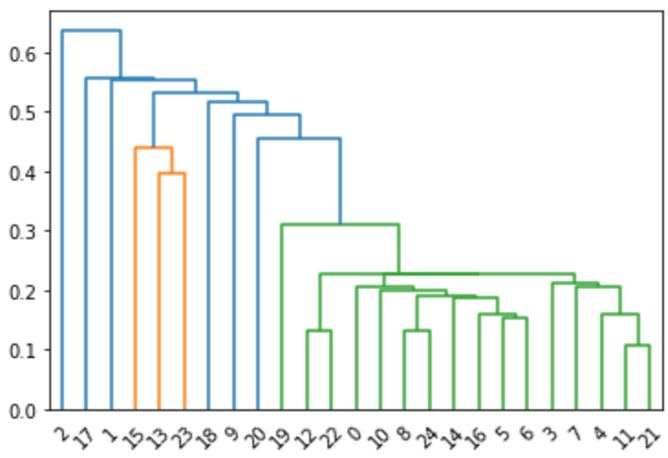
Out[94]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	-0.256098	-0.152778	-0.894737	-0.631579	-0.744856	-0.100000
1	0.000000	0.000000	-0.855263	-1.000000	0.000000	-0.533333
2	-0.378049	-0.527778	-0.407895	-0.842105	-0.702539	-0.833333
3	-0.256098	-0.333333	-0.868421	-0.684211	-0.584371	-0.300000
4	-0.329268	-0.236111	-0.750000	-0.631579	-0.760165	-0.233333

4.Dendogram

In [67]:
```
import  scipy.cluster.hierarchy  as sch
```

In [42]:
```
dendogram=sch.dendrogram(sch.linkage(data,method='single'))
plt.show()
```



5.Agglomerative Clustring

In [43]:
```
from sklearn.cluster import AgglomerativeClustering
```

In [44]:
```
hc=AgglomerativeClustering(n_clusters=4)
```

In [45]:
```
hc.fit(data)
```

Out[45]:
```
AgglomerativeClustering(n_clusters=4)
```

In [47]:
```
cluster=hc.fit_predict(data)
cluster
```

Out[47]:
```
array([3, 0, 2, 3, 3, 0, 0, 3, 0, 0, 0, 3, 3, 1, 0, 1, 0, 1, 2, 0, 2, 3,
       3, 1, 0], dtype=int64)
```

In [96]:
```
pd.DataFrame(cluster)
```

Out[96]:

	0
0	3
1	0
2	2
3	3
4	3
5	0
6	0
7	3
8	0
9	0
10	0
11	3
12	3
13	1
14	0
15	1
16	0
17	1
18	2
19	0
20	2
21	3
22	3
23	1
24	0