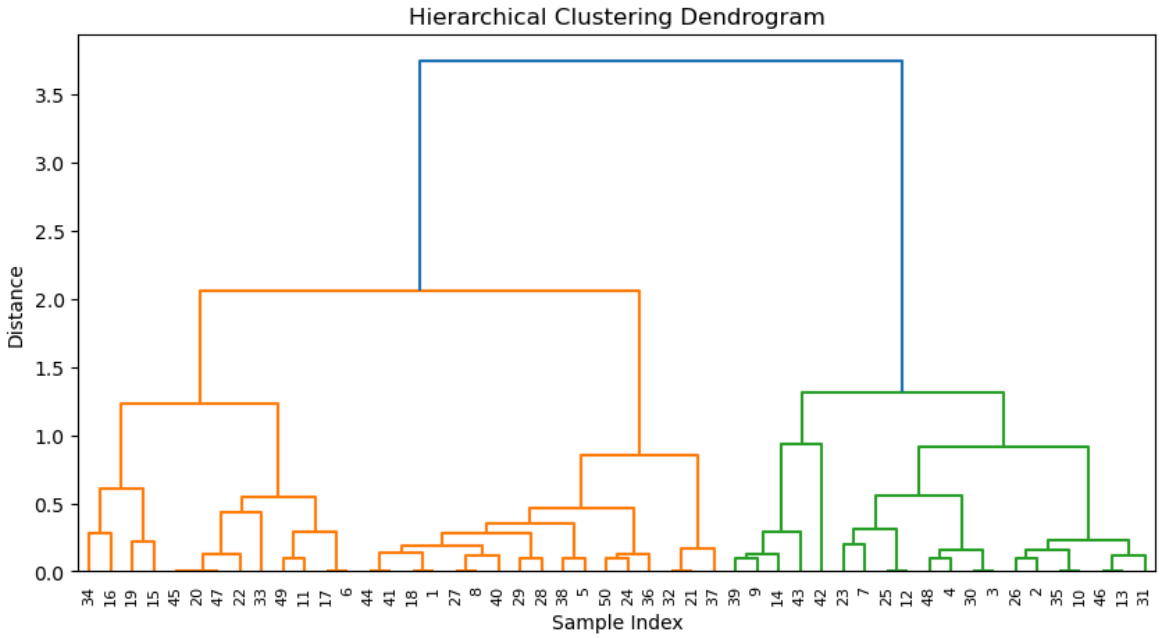
In [1]:
```
#now we disscused the heirarchical clustering.......
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.cluster.hierarchy import dendrogram, linkage

dataset = pd.read_csv('iris')

# Select only sepal_length and sepal_width for clustering
data = dataset[['sepal_length', 'sepal_width']].values[:50]

# Perform hierarchical clustering
linked = linkage(data, method='ward')

# Create a dendrogram
plt.figure(figsize=(10, 5))
dendrogram(linked, orientation='top', labels=range(1, 51), distance_sort=
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()

```



In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
from scipy.cluster.hierarchy import dendrogram,linkage
```

In [5]:
```
data=pd.read_csv('iris')
data
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [7]:
```
data.duplicated().sum()
dataset=data.drop_duplicates()
dataset
```

Out[7]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

149 rows × 5 columns

In [8]:
```
input=dataset.iloc[:,[2,3]].values[:50]
input
```

Out[8]:
```
array([[1.4, 0.2],
       [1.4, 0.2],
       [1.3, 0.2],
       [1.5, 0.2],
       [1.4, 0.2],
       [1.7, 0.4],
       [1.4, 0.3],
       [1.5, 0.2],
       [1.4, 0.2],
       [1.5, 0.1],
       [1.5, 0.2],
       [1.6, 0.2],
       [1.4, 0.1],
       [1.1, 0.1],
       [1.2, 0.2],
       [1.5, 0.4],
       [1.3, 0.4],
       [1.4, 0.3],
       [1.7, 0.3],
       [1.5, 0.3],
       [1.7, 0.2],
       [1.5, 0.4],
       [1. , 0.2],
       [1.7, 0.5],
       [1.9, 0.2],
       [1.6, 0.2],
       [1.6, 0.4],
       [1.5, 0.2],
       [1.4, 0.2],
       [1.6, 0.2],
       [1.6, 0.2],
       [1.5, 0.4],
       [1.5, 0.1],
       [1.4, 0.2],
       [1.5, 0.2],
       [1.2, 0.2],
       [1.3, 0.2],
       [1.4, 0.1],
       [1.3, 0.2],
       [1.5, 0.2],
       [1.3, 0.3],
       [1.3, 0.3],
       [1.3, 0.2],
       [1.6, 0.6],
       [1.9, 0.4],
       [1.4, 0.3],
       [1.6, 0.2],
       [1.4, 0.2],
       [1.5, 0.2],
       [1.4, 0.2]])
```

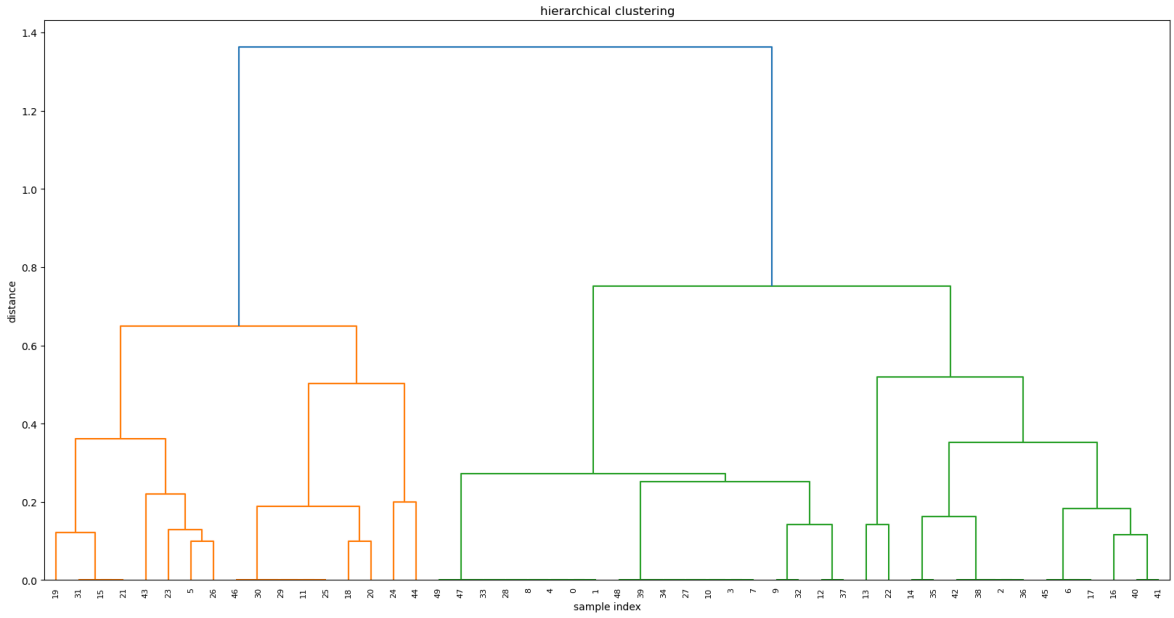
In [9]:
```
1 linked=linkage(input,method='ward')
```

In [12]:
```
1 linked[:50]
```

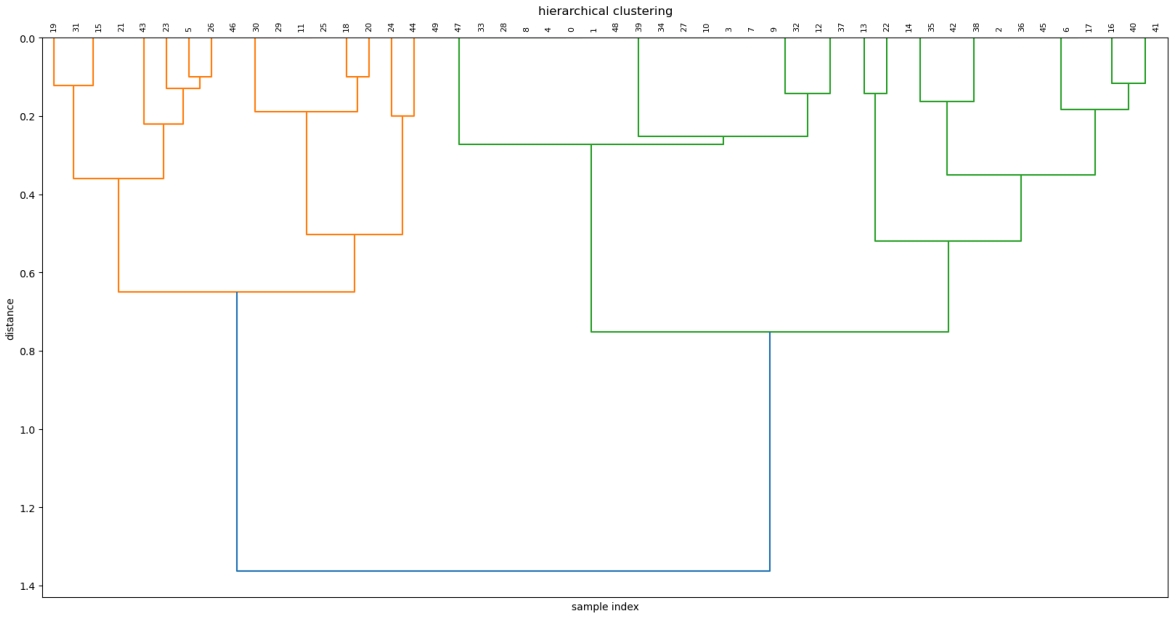
Out[12]:
```
array([[ 0.        ,  1.        ,  0.        ,  2.        ],
       [ 4.        , 50.        ,  0.        ,  3.        ],
       [ 2.        , 36.        ,  0.        ,  2.        ],
       [ 3.        ,  7.        ,  0.        ,  2.        ],
       [ 8.        , 51.        ,  0.        ,  4.        ],
       [ 6.        , 17.        ,  0.        ,  2.        ],
       [10.        , 53.        ,  0.        ,  3.        ],
       [28.        , 54.        ,  0.        ,  5.        ],
       [ 9.        , 32.        ,  0.        ,  2.        ],
       [27.        , 56.        ,  0.        ,  4.        ],
       [11.        , 25.        ,  0.        ,  2.        ],
       [12.        , 37.        ,  0.        ,  2.        ],
       [14.        , 35.        ,  0.        ,  2.        ],
       [15.        , 21.        ,  0.        ,  2.        ],
       [40.        , 41.        ,  0.        ,  2.        ],
       [45.        , 55.        ,  0.        ,  3.        ],
       [38.        , 52.        ,  0.        ,  3.        ],
       [42.        , 66.        ,  0.        ,  4.        ],
       [29.        , 60.        ,  0.        ,  3.        ],
       [30.        , 68.        ,  0.        ,  4.        ],
       [46.        , 69.        ,  0.        ,  5.        ],
       [34.        , 59.        ,  0.        ,  5.        ],
       [39.        , 71.        ,  0.        ,  6.        ],
       [48.        , 72.        ,  0.        ,  7.        ],
       [31.        , 63.        ,  0.        ,  3.        ],
       [33.        , 57.        ,  0.        ,  6.        ],
       [47.        , 75.        ,  0.        ,  7.        ],
       [49.        , 76.        ,  0.        ,  8.        ],
       [ 5.        , 26.        ,  0.1       ,  2.        ],
       [18.        , 20.        ,  0.1       ,  2.        ],
       [16.        , 64.        ,  0.11547005,  3.        ],
       [19.        , 74.        ,  0.12247449,  4.        ],
       [23.        , 78.        ,  0.12909944,  3.        ],
       [13.        , 22.        ,  0.14142136,  2.        ],
       [58.        , 61.        ,  0.14142136,  4.        ],
       [62.        , 67.        ,  0.16329932,  6.        ],
       [65.        , 80.        ,  0.18257419,  6.        ],
       [70.        , 79.        ,  0.18898224,  7.        ],
       [24.        , 44.        ,  0.2       ,  2.        ],
       [43.        , 82.        ,  0.21984843,  4.        ],
       [73.        , 84.        ,  0.25226249, 11.        ],
       [77.        , 90.        ,  0.2725039 , 19.        ],
       [85.        , 86.        ,  0.35118846, 12.        ],
       [81.        , 89.        ,  0.36055513,  8.        ],
       [87.        , 88.        ,  0.50205925,  9.        ],
       [83.        , 92.        ,  0.518698  , 14.        ],
       [93.        , 94.        ,  0.64888038, 17.        ],
       [91.        , 95.        ,  0.75158222, 33.        ],
       [96.        , 97.        ,  1.36260472, 50.        ]])
```

In [20]:

```
# dendrogram(linked, orientation='top', labels=range(1, 51), distance_sor
plt.figure(figsize=(20,10))
#this method is divisive hierarchical algorithm
dendrogram(linked,orientation='top',distance_sort='accending', show_leaf_
plt.title('hierarchical clustering')
plt.xlabel('sample index')
plt.ylabel('distance')

plt.show()
```



In [21]:
```
# dendrogram(linked, orientation='top', labels=range(1, 51), distance_sor
plt.figure(figsize=(20,10))
# this is agglomerative hierarchical clustering ...
dendrogram(linked,orientation='bottom',distance_sort='decending', show_le
plt.title('hierarchical clustering')
plt.xlabel('sample index')
plt.ylabel('distance')

plt.show()
```



In [ ]:
```
1
```