Implement hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method. Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data

In [37]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt
import numpy as np
from scipy.spatial.distance import cdist
```

In [38]:
```
# Load the dataset (adjust the encoding if needed)
data = pd.read_csv("C:/Users/Atharva/OneDrive/Desktop/LP3 code/sales_data_sample.csv", encoding='ISO-8859-1')
print(data.head())
```

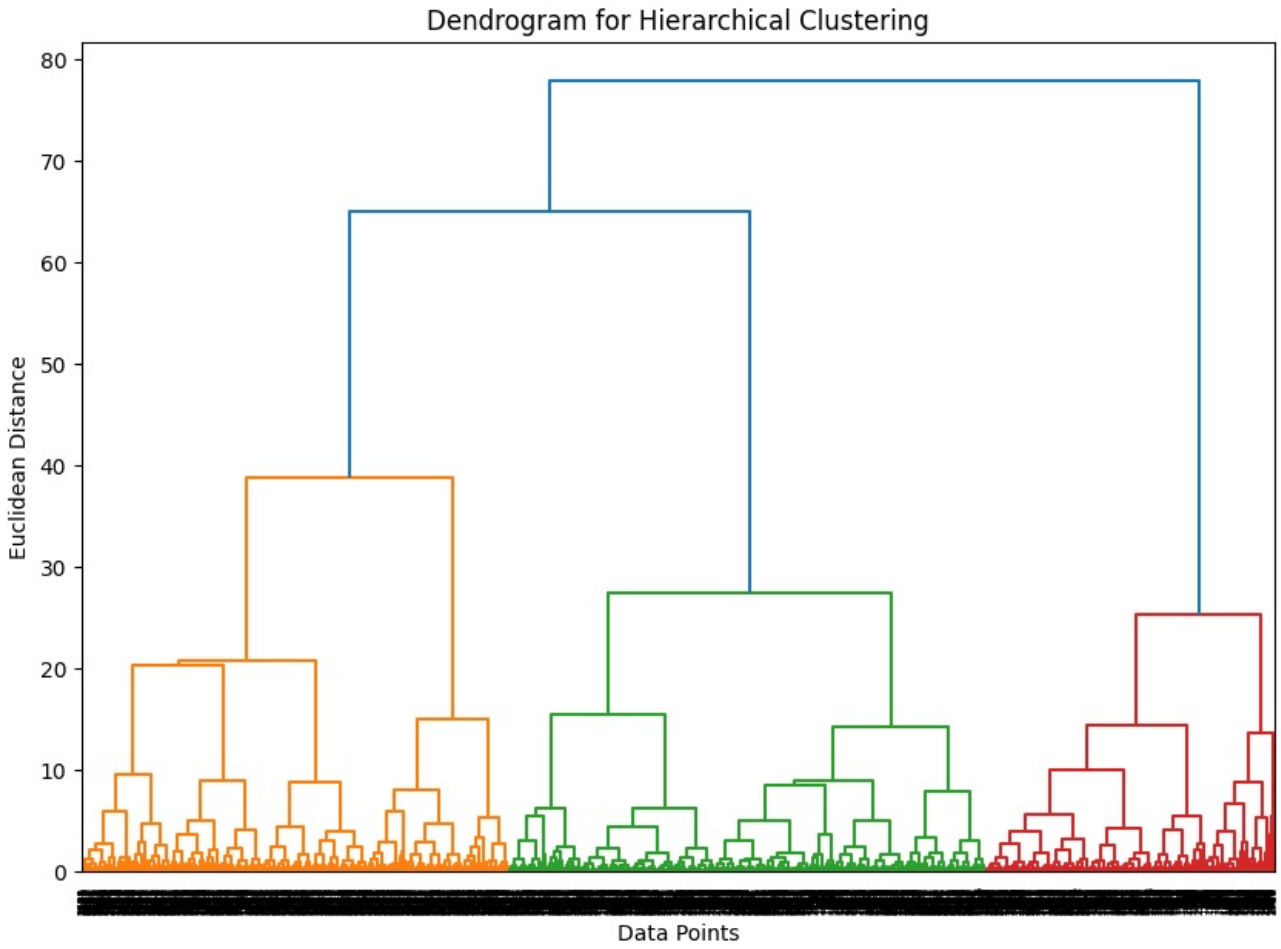
```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

          ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                   ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN             Yu             Kwai    Small
1      51100  France      EMEA        Henriot             Paul    Small
2      75508  France      EMEA       Da Cunha           Daniel   Medium
3      90003     USA       NaN          Young            Julie   Medium
4        NaN     USA       NaN          Brown            Julie   Medium

[5 rows x 25 columns]
```

In [39]:
```
# Select relevant numeric features for clustering
numeric_data = data[['QUANTITYORDERED', 'PRICEEACH', 'SALES']]
```

In [40]:
```
# Handle missing values (if any) by dropping rows with NaN values
numeric_data = numeric_data.dropna()
```

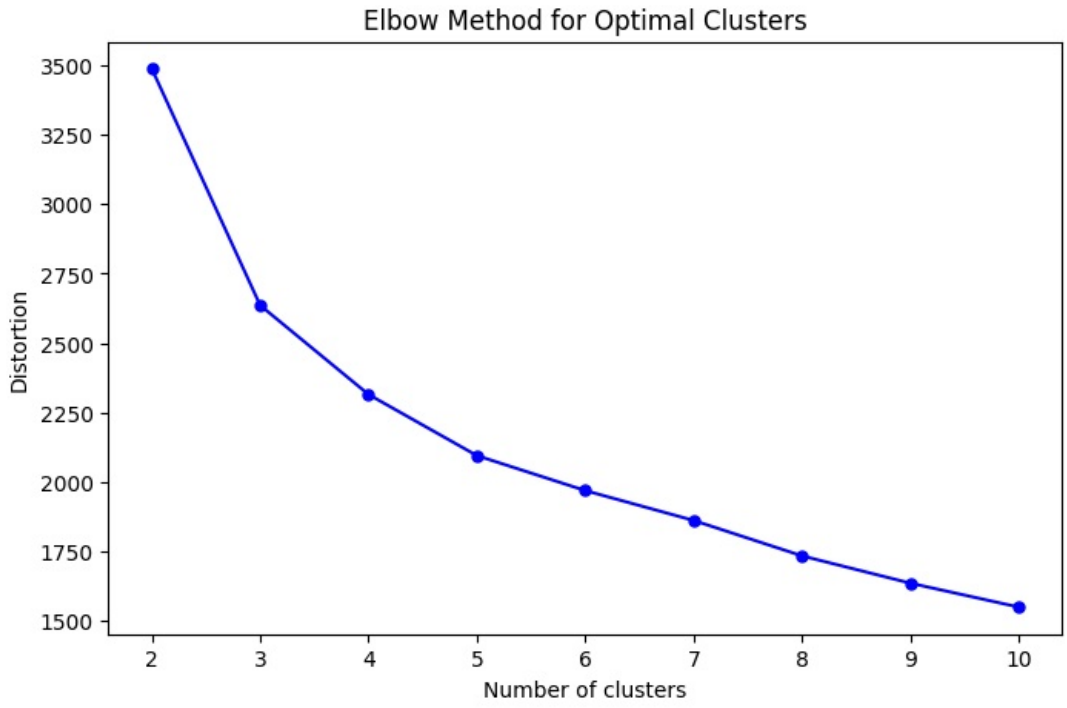
In [41]:
```
# Scale the features for hierarchical clustering
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_data)
```

In [42]:
```
# Perform hierarchical clustering and plot the dendrogram
linked = linkage(scaled_data, method='ward')
plt.figure(figsize=(10, 7))
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=False)
plt.title('Dendrogram for Hierarchical Clustering')
plt.xlabel('Data Points')
plt.ylabel('Euclidean Distance')
plt.show()
```



In [43]:
```
# Elbow method to determine the optimal number of clusters
distortions = []
K = range(2, 11)  # Start from 2 as 1 cluster distortion doesn't provide useful information
for k in K:
    model = AgglomerativeClustering(n_clusters=k, linkage='ward')
    model.fit(scaled_data)
    # Calculate the mean distance between each point and its assigned cluster's centroid
    labels = model.labels_
    centroids = [scaled_data[labels == i].mean(axis=0) for i in range(k)]
    distortion = sum(np.min(cdist(scaled_data, centroids, 'euclidean'), axis=1))
    distortions.append(distortion)
```

In [44]:
```
# Plot the elbow plot
plt.figure(figsize=(8, 5))
plt.plot(K, distortions, 'bo-', markersize=5)
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.title('Elbow Method for Optimal Clusters')
plt.show()
```



In [45]:
```
# Fit Agglomerative Clustering with the number of clusters determined (e.g., 3 based on elbow method observatio
optimal_k = 3  # Adjust this based on elbow plot observation if needed
hierarchical_cluster = AgglomerativeClustering(n_clusters=optimal_k, metric='euclidean', linkage='ward')
data['Cluster'] = hierarchical_cluster.fit_predict(scaled_data)
```

In [46]:
```
# Show the first few rows with cluster labels
print(data[['QUANTITYORDERED', 'PRICEEACH', 'SALES', 'Cluster']].head())
```

```
   QUANTITYORDERED  PRICEEACH    SALES  Cluster
0               30      95.70  2871.00        2
1               34      81.35  2765.90        2
2               41      94.74  3884.34        1
3               45      83.26  3746.70        0
4               49     100.00  5205.27        1
```

In [ ]: