# C3_W1_KMeans_Assignment

October 12, 2023

## 1 K-means Clustering

In this exercise, you will implement the K-means algorithm and use it for image compression.

- You will start with a sample dataset that will help you gain an intuition of how the K-means algorithm works.
- After that, you will use the K-means algorithm for image compression by reducing the number of colors that occur in an image to only those that are most common in that image.

## 2 Outline

- Section **??**
    - Section **??**
        - Section **??**
    - Section **??**
        - Section **??**
- Section **??**
- Section **??**
- Section **??**
    - Section **??**
    - Section **??**
    - Section **??**

***NOTE:*** *To prevent errors from the autograder, you are not allowed to edit or delete non-graded cells in this lab. Please also refrain from adding any new cells.* ***Once you have passed this assignment*** *and want to experiment with any of the non-graded code, you may follow the instructions at the bottom of this notebook.*

First, run the cell below to import the packages needed in this assignment:

- numpy is the fundamental package for scientific computing with Python.
- matplotlib is a popular library to plot graphs in Python.
- `utils.py` contains helper functions for this assignment. You do not need to modify code in this file.

[3]:

```
import numpy as np
import matplotlib.pyplot as plt
from utils import *
```

```
%matplotlib inline
```

## 1 - Implementing K-means

The K-means algorithm is a method to automatically cluster similar data points together.

- Concretely, you are given a training set $\{x^{(1)}, ..., x^{(m)}\}$, and you want to group the data into a few cohesive "clusters".

- K-means is an iterative procedure that

  - Starts by guessing the initial centroids, and then
  - Refines this guess by
    - Repeatedly assigning examples to their closest centroids, and then
    - Recomputing the centroids based on the assignments.

- In pseudocode, the K-means algorithm is as follows:

  ```
  # Initialize centroids
  # K is the number of clusters
  centroids = kMeans_init_centroids(X, K)

  for iter in range(iterations):
      # Cluster assignment step:
      # Assign each data point to the closest centroid.
      # idx[i] corresponds to the index of the centroid
      # assigned to example i
      idx = find_closest_centroids(X, centroids)

      # Move centroid step:
      # Compute means based on centroid assignments
      centroids = compute_centroids(X, idx, K)
  ```

- The inner-loop of the algorithm repeatedly carries out two steps:

  1. Assigning each training example $x^{(i)}$ to its closest centroid, and
  2. Recomputing the mean of each centroid using the points assigned to it.

- The $K$-means algorithm will always converge to some final set of means for the centroids.

- However, the converged solution may not always be ideal and depends on the initial setting of the centroids.

  - Therefore, in practice the K-means algorithm is usually run a few times with different random initializations.
  - One way to choose between these different solutions from different random initializations is to choose the one with the lowest cost function value (distortion).

You will implement the two phases of the K-means algorithm separately in the next sections. * You will start by completing `find_closest_centroid` and then proceed to complete `compute_centroids`.

### 1.1 Finding closest centroids

In the “cluster assignment” phase of the K-means algorithm, the algorithm assigns every training example $x^{(i)}$ to its closest centroid, given the current positions of centroids.

### Exercise 1

Your task is to complete the code in `find_closest_centroids`. * This function takes the data matrix `X` and the locations of all centroids inside `centroids` * It should output a one-dimensional array `idx` (which has the same number of elements as `X`) that holds the index of the closest centroid (a value in $\{0, ..., K-1\}$, where $K$ is total number of centroids) to every training example . *(Note: The index range 0 to K-1 varies slightly from what is shown in the lectures (i.e. 1 to K) because Python list indices start at 0 instead of 1)* * Specifically, for every example $x^{(i)}$ we set

$$c^{(i)} := j \quad \text{that minimizes} \quad ||x^{(i)} - \mu_j||^2,$$

where * $c^{(i)}$ is the index of the centroid that is closest to $x^{(i)}$ (corresponds to `idx[i]` in the starter code), and * $\mu_j$ is the position (value) of the $j$'th centroid. (stored in `centroids` in the starter code) * $||x^{(i)} - \mu_j||$ is the L2-norm

If you get stuck, you can check out the hints presented after the cell below to help you with the implementation.

[12]:

```
# UNQ_C1
# GRADED FUNCTION: find_closest_centroids

def find_closest_centroids(X, centroids):
    """
    Computes the centroid memberships for every example

    Args:
        X (ndarray): (m, n) Input values
        centroids (ndarray): (K, n) centroids

    Returns:
        idx (array_like): (m,) closest centroids

    """

    # Set K
    K = centroids.shape[0]


    # You need to return the following variables correctly
    idx = np.zeros(X.shape[0], dtype=int)

    ### START CODE HERE ###
    for i in range(X.shape[0]):
        distance = []
        for j in range(centroids.shape[0]):

            norm_ij = np.linalg.norm(X[i]-centroids[j])
```

```
#             print(f"Norm_ij = norm({X[i]}-{centroids[j]}) = {norm_ij}")
            distance.append(norm_ij)
#         print(f"distance: {distance}")
        idx[i] = np.argmin(distance)
#         print(f"idx[{i}] = {idx[i]}")
#     print(idx)



     ### END CODE HERE ###

    return idx
```

Click for hints

- Here's how you can structure the overall implementation for this function

```
def find_closest_centroids(X, centroids):

    # Set K
    K = centroids.shape[0]

    # You need to return the following variables correctly
    idx = np.zeros(X.shape[0], dtype=int)

    ### START CODE HERE ###
    for i in range(X.shape[0]):
        # Array to hold distance between X[i] and each centroids[j]
        distance = []
        for j in range(centroids.shape[0]):
            norm_ij = # Your code to calculate the norm between (X[i] - centroids[j])
            distance.append(norm_ij)

        idx[i] = # Your code here to calculate index of minimum value in distance
    ### END CODE HERE ###
    return idx
```

- If you're still stuck, you can check the hints presented below to figure out how to calculate `norm_ij` and `idx[i]`.

  Hint to calculate norm_ij You can use np.linalg.norm to calculate the norm

  More hints to calculate norm_ij You can compute norm_ij as norm_ij = np.linalg.norm(X[i] - centroids[j])

  Hint to calculate idx[i] You can use np.argmin to find the index of the minimum value

  More hints to calculate idx[i] You can compute idx[i] as idx[i] = np.argmin(distance)

Now let's check your implementation using an example dataset

[5]:

```
# Load an example dataset that we will be using
X = load_data()
```

The code below prints the first five elements in the variable X and the dimensions of the variable

[6]:

```
print("First five elements of X are:\n", X[:5])
print('The shape of X is:', X.shape)
```

```
First five elements of X are:
 [[1.84207953 4.6075716 ]
 [5.65858312 4.79996405]
 [6.35257892 3.2908545 ]
 [2.90401653 4.61220411]
 [3.23197916 4.93989405]]
The shape of X is: (300, 2)
```

[7]:

```
# Select an initial set of centroids (3 Centroids)
initial_centroids = np.array([[3,3], [6,2], [8,5]])

# Find closest centroids using initial_centroids
idx = find_closest_centroids(X, initial_centroids)

# Print closest centroids for the first three elements
print("First three elements in idx are:", idx[:3])

# UNIT TEST
from public_tests import *

find_closest_centroids_test(find_closest_centroids)
```

```
Norm_ij = norm([1.84207953 4.6075716 ]-[3 3]) = 1.981178002041907
Norm_ij = norm([1.84207953 4.6075716 ]-[6 2]) = 4.907925457663594
Norm_ij = norm([1.84207953 4.6075716 ]-[8 5]) = 6.170412024053506
distance: [1.981178002041907, 4.907925457663594, 6.170412024053506]
idx[0] = 0
Norm_ij = norm([5.65858312 4.79996405]-[3 3]) = 3.2105972663230307
Norm_ij = norm([5.65858312 4.79996405]-[6 2]) = 2.820702783295523
Norm_ij = norm([5.65858312 4.79996405]-[8 5]) = 2.3499462509952416
distance: [3.2105972663230307, 2.820702783295523, 2.3499462509952416]
idx[1] = 2
Norm_ij = norm([6.35257892 3.2908545 ]-[3 3]) = 3.365171876090534
Norm_ij = norm([6.35257892 3.2908545 ]-[6 2]) = 1.33813946654494
Norm_ij = norm([6.35257892 3.2908545 ]-[8 5]) = 2.373852261323399
distance: [3.365171876090534, 1.33813946654494, 2.373852261323399]
idx[2] = 1
Norm_ij = norm([2.90401653 4.61220411]-[3 3]) = 1.6150588004146713
Norm_ij = norm([2.90401653 4.61220411]-[6 2]) = 4.050768322520906
Norm_ij = norm([2.90401653 4.61220411]-[8 5]) = 5.1107174862951465
```

```
distance: [1.6150588004146713, 4.050768322520906, 5.1107174862951465]
idx[3] = 0
Norm_ij = norm([3.23197916 4.93989405]-[3 3]) = 1.9537152467976249
Norm_ij = norm([3.23197916 4.93989405]-[6 2]) = 4.0379346735648705
Norm_ij = norm([3.23197916 4.93989405]-[8 5]) = 4.768399677270132
distance: [1.9537152467976249, 4.0379346735648705, 4.768399677270132]
idx[4] = 0
Norm_ij = norm([1.24792268 4.93267846]-[3 3]) = 2.6086435107963104
Norm_ij = norm([1.24792268 4.93267846]-[6 2]) = 5.584159901758319
Norm_ij = norm([1.24792268 4.93267846]-[8 5]) = 6.752412926431547
distance: [2.6086435107963104, 5.584159901758319, 6.752412926431547]
idx[5] = 0
Norm_ij = norm([1.97619886 4.43489674]-[3 3]) = 1.7626960691610503
Norm_ij = norm([1.97619886 4.43489674]-[6 2]) = 4.703158273676196
Norm_ij = norm([1.97619886 4.43489674]-[8 5]) = 6.050249734222309
distance: [1.7626960691610503, 4.703158273676196, 6.050249734222309]
idx[6] = 0
Norm_ij = norm([2.23454135 5.05547168]-[3 3]) = 2.1933742892246193
Norm_ij = norm([2.23454135 5.05547168]-[6 2]) = 4.849184056681204
Norm_ij = norm([2.23454135 5.05547168]-[8 5]) = 5.765725497657202
distance: [2.1933742892246193, 4.849184056681204, 5.765725497657202]
idx[7] = 0
Norm_ij = norm([2.98340757 4.84046406]-[3 3]) = 1.8405388530112585
Norm_ij = norm([2.98340757 4.84046406]-[6 2]) = 4.143436491849299
Norm_ij = norm([2.98340757 4.84046406]-[8 5]) = 5.0191285406757205
distance: [1.8405388530112585, 4.143436491849299, 5.0191285406757205]
idx[8] = 0
Norm_ij = norm([2.97970391 4.80671081]-[3 3]) = 1.806824804758238
Norm_ij = norm([2.97970391 4.80671081]-[6 2]) = 4.123083075746503
Norm_ij = norm([2.97970391 4.80671081]-[8 5]) = 5.024015682089771
distance: [1.806824804758238, 4.123083075746503, 5.024015682089771]
idx[9] = 0
Norm_ij = norm([2.11496411 5.37373587]-[3 3]) = 2.533359524908645
Norm_ij = norm([2.11496411 5.37373587]-[6 2]) = 5.145444348668002
Norm_ij = norm([2.11496411 5.37373587]-[8 5]) = 5.896891207096016
distance: [2.533359524908645, 5.145444348668002, 5.896891207096016]
idx[10] = 0
Norm_ij = norm([2.12169543 5.20854212]-[3 3]) = 2.376778743196578
Norm_ij = norm([2.12169543 5.20854212]-[6 2]) = 5.03348674837871
Norm_ij = norm([2.12169543 5.20854212]-[8 5]) = 5.882002586992182
distance: [2.376778743196578, 5.03348674837871, 5.882002586992182]
idx[11] = 0
Norm_ij = norm([1.5143529  4.77003303]-[3 3]) = 2.3108795367072363
Norm_ij = norm([1.5143529  4.77003303]-[6 2]) = 5.272012223034195
Norm_ij = norm([1.5143529  4.77003303]-[8 5]) = 6.489722880947639
distance: [2.3108795367072363, 5.272012223034195, 6.489722880947639]
idx[12] = 0
Norm_ij = norm([2.16979227 5.27435354]-[3 3]) = 2.421142063299345
Norm_ij = norm([2.16979227 5.27435354]-[6 2]) = 5.0390358543506135
Norm_ij = norm([2.16979227 5.27435354]-[8 5]) = 5.836659320829771
distance: [2.421142063299345, 5.0390358543506135, 5.836659320829771]
idx[13] = 0
Norm_ij = norm([0.41852373 4.88312522]-[3 3]) = 3.195337276086454
Norm_ij = norm([0.41852373 4.88312522]-[6 2]) = 6.282140427433427
Norm_ij = norm([0.41852373 4.88312522]-[8 5]) = 7.582377075174331
distance: [3.195337276086454, 6.282140427433427, 7.582377075174331]
idx[14] = 0
Norm_ij = norm([2.47053695 4.80418944]-[3 3]) = 1.8802740921707952
Norm_ij = norm([2.47053695 4.80418944]-[6 2]) = 4.507836271878243
Norm_ij = norm([2.47053695 4.80418944]-[8 5]) = 5.532929008765773
distance: [1.8802740921707952, 4.507836271878243, 5.532929008765773]
idx[15] = 0
Norm_ij = norm([4.06069132 4.99503862]-[3 3]) = 2.259478962929985
Norm_ij = norm([4.06069132 4.99503862]-[6 2]) = 3.5680771402736555
Norm_ij = norm([4.06069132 4.99503862]-[8 5]) = 3.9393117999040004
distance: [2.259478962929985, 3.5680771402736555, 3.9393117999040004]
idx[16] = 0
Norm_ij = norm([3.00708934 4.67897758]-[3 3]) = 1.678992544368812
Norm_ij = norm([3.00708934 4.67897758]-[6 2]) = 4.016769235515177
Norm_ij = norm([3.00708934 4.67897758]-[8 5]) = 5.003220190588045
distance: [1.678992544368812, 4.016769235515177, 5.003220190588045]
idx[17] = 0
Norm_ij = norm([0.66632346 4.87187949]-[3 3]) = 2.9916515570751643
Norm_ij = norm([0.66632346 4.87187949]-[6 2]) = 6.057705612206054
Norm_ij = norm([0.66632346 4.87187949]-[8 5]) = 7.334795601643093
distance: [2.9916515570751643, 6.057705612206054, 7.334795601643093]
idx[18] = 0
Norm_ij = norm([3.1621865  4.83658301]-[3 3]) = 1.8437303487275412
Norm_ij = norm([3.1621865  4.83658301]-[6 2]) = 4.012404340573574
Norm_ij = norm([3.1621865  4.83658301]-[8 5]) = 4.840572749748981
distance: [1.8437303487275412, 4.012404340573574, 4.840572749748981]
idx[19] = 0
Norm_ij = norm([0.51155258 4.91052923]-[3 3]) = 3.1372746926132398
Norm_ij = norm([0.51155258 4.91052923]-[6 2]) = 6.21242589280012
Norm_ij = norm([0.51155258 4.91052923]-[8 5]) = 7.488981891747967
distance: [3.1372746926132398, 6.21242589280012, 7.488981891747967]
idx[20] = 0
Norm_ij = norm([3.1342801  4.96178114]-[3 3]) = 1.9663713746582776
Norm_ij = norm([3.1342801  4.96178114]-[6 2]) = 4.121225311000468
Norm_ij = norm([3.1342801  4.96178114]-[8 5]) = 4.865869997325887
distance: [1.9663713746582776, 4.121225311000468, 4.865869997325887]
idx[21] = 0
Norm_ij = norm([2.04974595 5.6241395 ]-[3 3]) = 2.7908942749254173
Norm_ij = norm([2.04974595 5.6241395 ]-[6 2]) = 5.360866921064424
Norm_ij = norm([2.04974595 5.6241395 ]-[8 5]) = 5.982898408727239
distance: [2.7908942749254173, 5.360866921064424, 5.982898408727239]
```

```
idx[22] = 0
Norm_ij = norm([0.66582785 5.24399257]-[3 3]) = 3.2378792890501034
Norm_ij = norm([0.66582785 5.24399257]-[6 2]) = 6.243146668596056
Norm_ij = norm([0.66582785 5.24399257]-[8 5]) = 7.338229586683384
distance: [3.2378792890501034, 6.243146668596056, 7.338229586683384]
idx[23] = 0
Norm_ij = norm([1.01732013 4.84473647]-[3 3]) = 2.7081492064158765
Norm_ij = norm([1.01732013 4.84473647]-[6 2]) = 5.737562575486236
Norm_ij = norm([1.01732013 4.84473647]-[8 5]) = 6.98440584367393
distance: [2.7081492064158765, 5.737562575486236, 6.98440584367393]
idx[24] = 0
Norm_ij = norm([2.17893568 5.29758701]-[3 3]) = 2.4398878476594916
Norm_ij = norm([2.17893568 5.29758701]-[6 2]) = 5.04723812384817
Norm_ij = norm([2.17893568 5.29758701]-[8 5]) = 5.828666047351037
distance: [2.4398878476594916, 5.04723812384817, 5.828666047351037]
idx[25] = 0
Norm_ij = norm([2.85962615 5.26041997]-[3 3]) = 2.2647744421148857
Norm_ij = norm([2.85962615 5.26041997]-[6 2]) = 4.526840655722172
Norm_ij = norm([2.85962615 5.26041997]-[8 5]) = 5.146966279633928
distance: [2.2647744421148857, 4.526840655722172, 5.146966279633928]
idx[26] = 0
Norm_ij = norm([1.30882588 5.30158701]-[3 3]) = 2.856111456587242
Norm_ij = norm([1.30882588 5.30158701]-[6 2]) = 5.73651386898476
Norm_ij = norm([1.30882588 5.30158701]-[8 5]) = 6.69796728717336
distance: [2.856111456587242, 5.73651386898476, 6.69796728717336]
idx[27] = 0
Norm_ij = norm([0.99253246 5.01567424]-[3 3]) = 2.844796754510685
Norm_ij = norm([0.99253246 5.01567424]-[6 2]) = 5.845427470849198
Norm_ij = norm([0.99253246 5.01567424]-[8 5]) = 7.007485074526588
distance: [2.844796754510685, 5.845427470849198, 7.007485074526588]
idx[28] = 0
Norm_ij = norm([1.40372638 4.57527684]-[3 3]) = 2.2426739845339583
Norm_ij = norm([1.40372638 4.57527684]-[6 2]) = 5.268565459342714
Norm_ij = norm([1.40372638 4.57527684]-[8 5]) = 6.609933086358191
distance: [2.2426739845339583, 5.268565459342714, 6.609933086358191]
idx[29] = 0
Norm_ij = norm([2.66046572 5.19623848]-[3 3]) = 2.2223291788508077
Norm_ij = norm([2.66046572 5.19623848]-[6 2]) = 4.622599873561734
Norm_ij = norm([2.66046572 5.19623848]-[8 5]) = 5.343139135641352
distance: [2.2223291788508077, 4.622599873561734, 5.343139135641352]
idx[30] = 0
Norm_ij = norm([2.79995882 5.11526323]-[3 3]) = 2.124701153973224
Norm_ij = norm([2.79995882 5.11526323]-[6 2]) = 4.465996924767976
Norm_ij = norm([2.79995882 5.11526323]-[8 5]) = 5.201318476362819
distance: [2.124701153973224, 4.465996924767976, 5.201318476362819]
idx[31] = 0
Norm_ij = norm([2.06995345 4.6846713 ]-[3 3]) = 1.9243450778787836
Norm_ij = norm([2.06995345 4.6846713 ]-[6 2]) = 4.759487983291296
Norm_ij = norm([2.06995345 4.6846713 ]-[8 5]) = 5.938424389312099
distance: [1.9243450778787836, 4.759487983291296, 5.938424389312099]
idx[32] = 0
Norm_ij = norm([3.29765181 5.59205535]-[3 3]) = 2.609089407613827
Norm_ij = norm([3.29765181 5.59205535]-[6 2]) = 4.495058106603733
Norm_ij = norm([3.29765181 5.59205535]-[8 5]) = 4.739473393219792
distance: [2.609089407613827, 4.495058106603733, 4.739473393219792]
idx[33] = 0
Norm_ij = norm([1.8929766  4.89043209]-[3 3]) = 2.190715476813697
Norm_ij = norm([1.8929766  4.89043209]-[6 2]) = 5.022174718528255
Norm_ij = norm([1.8929766  4.89043209]-[8 5]) = 6.108006220244764
distance: [2.190715476813697, 5.022174718528255, 6.108006220244764]
idx[34] = 0
Norm_ij = norm([2.55983064 5.26397756]-[3 3]) = 2.3063701951129776
Norm_ij = norm([2.55983064 5.26397756]-[6 2]) = 4.742184598635292
Norm_ij = norm([2.55983064 5.26397756]-[8 5]) = 5.446570189027362
distance: [2.3063701951129776, 4.742184598635292, 5.446570189027362]
idx[35] = 0
Norm_ij = norm([1.15354031 4.67866717]-[3 3]) = 2.495463255863491
Norm_ij = norm([1.15354031 4.67866717]-[6 2]) = 5.537456937769037
Norm_ij = norm([1.15354031 4.67866717]-[8 5]) = 6.8539962827069365
distance: [2.495463255863491, 5.537456937769037, 6.8539962827069365]
idx[36] = 0
Norm_ij = norm([2.25150754 5.4450031 ]-[3 3]) = 2.5570062836259457
Norm_ij = norm([2.25150754 5.4450031 ]-[6 2]) = 5.091094392048579
Norm_ij = norm([2.25150754 5.4450031 ]-[8 5]) = 5.765691055092864
distance: [2.5570062836259457, 5.091094392048579, 5.765691055092864]
idx[37] = 0
Norm_ij = norm([2.20960296 4.91469264]-[3 3]) = 2.0714186925514095
Norm_ij = norm([2.20960296 4.91469264]-[6 2]) = 4.781479159452961
Norm_ij = norm([2.20960296 4.91469264]-[8 5]) = 5.7910254086175925
distance: [2.0714186925514095, 4.781479159452961, 5.7910254086175925]
idx[38] = 0
Norm_ij = norm([1.59141937 4.83212573]-[3 3]) = 2.3110136443541958
Norm_ij = norm([1.59141937 4.83212573]-[6 2]) = 5.239896877508727
Norm_ij = norm([1.59141937 4.83212573]-[8 5]) = 6.41077900413804
distance: [2.3110136443541958, 5.239896877508727, 6.41077900413804]
idx[39] = 0
Norm_ij = norm([1.67838038 5.26903822]-[3 3]) = 2.625873733226457
Norm_ij = norm([1.67838038 5.26903822]-[6 2]) = 5.418764346382817
Norm_ij = norm([1.67838038 5.26903822]-[8 5]) = 6.327341956353284
distance: [2.625873733226457, 5.418764346382817, 6.327341956353284]
idx[40] = 0
Norm_ij = norm([2.59148642 4.92593394]-[3 3]) = 1.9687825860360617
Norm_ij = norm([2.59148642 4.92593394]-[6 2]) = 4.492110221138994
Norm_ij = norm([2.59148642 4.92593394]-[8 5]) = 5.409020702251052
distance: [1.9687825860360617, 4.492110221138994, 5.409020702251052]
idx[41] = 0
```

```
Norm_ij = norm([2.80996442 5.53849899]-[3 3]) = 2.5456022158926572
Norm_ij = norm([2.80996442 5.53849899]-[6 2]) = 4.764168561980675
Norm_ij = norm([2.80996442 5.53849899]-[8 5]) = 5.217897130006757
distance: [2.5456022158926572, 4.764168561980675, 5.217897130006757]
idx[42] = 0
Norm_ij = norm([0.95311627 5.58037108]-[3 3]) = 3.2936374852445085
Norm_ij = norm([0.95311627 5.58037108]-[6 2]) = 6.187898868234984
Norm_ij = norm([0.95311627 5.58037108]-[8 5]) = 7.070742596163362
distance: [3.2936374852445085, 6.187898868234984, 7.070742596163362]
idx[43] = 0
Norm_ij = norm([1.51775276 5.03836638]-[3 3]) = 2.520316323315795
Norm_ij = norm([1.51775276 5.03836638]-[6 2]) = 5.414998667461969
Norm_ij = norm([1.51775276 5.03836638]-[8 5]) = 6.482360779119623
distance: [2.520316323315795, 5.414998667461969, 6.482360779119623]
idx[44] = 0
Norm_ij = norm([3.23114248 5.78429665]-[3 3]) = 2.793874488496259
Norm_ij = norm([3.23114248 5.78429665]-[6 2]) = 4.689080196601555
Norm_ij = norm([3.23114248 5.78429665]-[8 5]) = 4.8329207843555615
distance: [2.793874488496259, 4.689080196601555, 4.8329207843555615]
idx[45] = 0
Norm_ij = norm([2.54180011 4.81098738]-[3 3]) = 1.8680531088240278
Norm_ij = norm([2.54180011 4.81098738]-[6 2]) = 4.456545358689915
Norm_ij = norm([2.54180011 4.81098738]-[8 5]) = 5.46147158356306
distance: [1.8680531088240278, 4.456545358689915, 5.46147158356306]
idx[46] = 0
Norm_ij = norm([3.81422865 4.73526796]-[3 3]) = 1.9168002534925437
Norm_ij = norm([3.81422865 4.73526796]-[6 2]) = 3.501326493639894
Norm_ij = norm([3.81422865 4.73526796]-[8 5]) = 4.194134571970357
distance: [1.9168002534925437, 3.501326493639894, 4.194134571970357]
idx[47] = 0
Norm_ij = norm([1.68495829 4.59643553]-[3 3]) = 2.0683184265716528
Norm_ij = norm([1.68495829 4.59643553]-[6 2]) = 5.035976810885425
Norm_ij = norm([1.68495829 4.59643553]-[8 5]) = 6.327923519642384
distance: [2.0683184265716528, 5.035976810885425, 6.327923519642384]
idx[48] = 0
Norm_ij = norm([2.17777173 4.86154019]-[3 3]) = 2.035040833843501
Norm_ij = norm([2.17777173 4.86154019]-[6 2]) = 4.774708490637411
Norm_ij = norm([2.17777173 4.86154019]-[8 5]) = 5.823874406696827
distance: [2.035040833843501, 4.774708490637411, 5.823874406696827]
idx[49] = 0
Norm_ij = norm([1.8173328  5.13333907]-[3 3]) = 2.439228831053435
Norm_ij = norm([1.8173328  5.13333907]-[6 2]) = 5.226138024229963
Norm_ij = norm([1.8173328  5.13333907]-[8 5]) = 6.184104871548888
distance: [2.439228831053435, 5.226138024229963, 6.184104871548888]
idx[50] = 0
Norm_ij = norm([1.85776553 4.86962414]-[3 3]) = 2.1909344997611977
Norm_ij = norm([1.85776553 4.86962414]-[6 2]) = 5.0391317750834075
Norm_ij = norm([1.85776553 4.86962414]-[8 5]) = 6.143617997260003
```

```
distance: [2.1909344997611977, 5.0391317750834075, 6.143617997260003]
idx[51] = 0
Norm_ij = norm([3.03084301 5.24057582]-[3 3]) = 2.240788098482533
Norm_ij = norm([3.03084301 5.24057582]-[6 2]) = 4.395136505138457
Norm_ij = norm([3.03084301 5.24057582]-[8 5]) = 4.974977179843914
distance: [2.240788098482533, 4.395136505138457, 4.974977179843914]
idx[52] = 0
Norm_ij = norm([2.92658295 5.09667923]-[3 3]) = 2.09796421610696
Norm_ij = norm([2.92658295 5.09667923]-[6 2]) = 4.3629479273811125
Norm_ij = norm([2.92658295 5.09667923]-[8 5]) = 5.074338129140522
distance: [2.09796421610696, 4.3629479273811125, 5.074338129140522]
idx[53] = 0
Norm_ij = norm([3.43493543 5.34080741]-[3 3]) = 2.380871304691701
Norm_ij = norm([3.43493543 5.34080741]-[6 2]) = 4.211953276571071
Norm_ij = norm([3.43493543 5.34080741]-[8 5]) = 4.5777684744872875
distance: [2.380871304691701, 4.211953276571071, 4.5777684744872875]
idx[54] = 0
Norm_ij = norm([3.20367116 4.85924759]-[3 3]) = 1.8703698913565492
Norm_ij = norm([3.20367116 4.85924759]-[6 2]) = 3.99934391239516
Norm_ij = norm([3.20367116 4.85924759]-[8 5]) = 4.798393643806516
distance: [1.8703698913565492, 3.99934391239516, 4.798393643806516]
idx[55] = 0
Norm_ij = norm([0.10511804 4.72916344]-[3 3]) = 3.3719946266828726
Norm_ij = norm([0.10511804 4.72916344]-[6 2]) = 6.495996182460992
Norm_ij = norm([0.10511804 4.72916344]-[8 5]) = 7.8995261638875425
distance: [3.3719946266828726, 6.495996182460992, 7.8995261638875425]
idx[56] = 0
Norm_ij = norm([1.40597916 5.06636822]-[3 3]) = 2.6097471276720965
Norm_ij = norm([1.40597916 5.06636822]-[6 2]) = 5.5233722980623154
Norm_ij = norm([1.40597916 5.06636822]-[8 5]) = 6.5943548237439025
distance: [2.6097471276720965, 5.5233722980623154, 6.5943548237439025]
idx[57] = 0
Norm_ij = norm([2.24185052 4.9244617 ]-[3 3]) = 2.0684156943223484
Norm_ij = norm([2.24185052 4.9244617 ]-[6 2]) = 4.76194957932122
Norm_ij = norm([2.24185052 4.9244617 ]-[8 5]) = 5.758644937824804
distance: [2.0684156943223484, 4.76194957932122, 5.758644937824804]
idx[58] = 0
Norm_ij = norm([1.36678395 5.26161095]-[3 3]) = 2.7896735943039874
Norm_ij = norm([1.36678395 5.26161095]-[6 2]) = 5.666109507804705
Norm_ij = norm([1.36678395 5.26161095]-[8 5]) = 6.6383729499737765
distance: [2.7896735943039874, 5.666109507804705, 6.6383729499737765]
idx[59] = 0
Norm_ij = norm([1.70725482 4.04231479]-[3 3]) = 1.6606053796440288
Norm_ij = norm([1.70725482 4.04231479]-[6 2]) = 4.753810144095388
Norm_ij = norm([1.70725482 4.04231479]-[8 5]) = 6.365202497431042
distance: [1.6606053796440288, 4.753810144095388, 6.365202497431042]
idx[60] = 0
Norm_ij = norm([1.91909566 5.57848447]-[3 3]) = 2.7958784542316844
Norm_ij = norm([1.91909566 5.57848447]-[6 2]) = 5.427645097756066
Norm_ij = norm([1.91909566 5.57848447]-[8 5]) = 6.108358362221452
distance: [2.7958784542316844, 5.427645097756066, 6.108358362221452]
idx[61] = 0
Norm_ij = norm([1.60156731 4.64453012]-[3 3]) = 2.1587249204146066
Norm_ij = norm([1.60156731 4.64453012]-[6 2]) = 5.132226576747843
Norm_ij = norm([1.60156731 4.64453012]-[8 5]) = 6.4082992819141875
distance: [2.1587249204146066, 5.132226576747843, 6.4082992819141875]
idx[62] = 0
Norm_ij = norm([0.37963437 5.26194729]-[3 3]) = 3.4616067928995045
Norm_ij = norm([0.37963437 5.26194729]-[6 2]) = 6.498369791587976
Norm_ij = norm([0.37963437 5.26194729]-[8 5]) = 7.62486647082081
distance: [3.4616067928995045, 6.498369791587976, 7.62486647082081]
idx[63] = 0
Norm_ij = norm([2.02134502 4.41267445]-[3 3]) = 1.7185501624551813
Norm_ij = norm([2.02134502 4.41267445]-[6 2]) = 4.653030562256329
Norm_ij = norm([2.02134502 4.41267445]-[8 5]) = 6.007434279341037
distance: [1.7185501624551813, 4.653030562256329, 6.007434279341037]
idx[64] = 0
Norm_ij = norm([1.12036737 5.20880747]-[3 3]) = 2.9003188177984125
Norm_ij = norm([1.12036737 5.20880747]-[6 2]) = 5.840142116769111
Norm_ij = norm([1.12036737 5.20880747]-[8 5]) = 6.882800712110645
distance: [2.9003188177984125, 5.840142116769111, 6.882800712110645]
idx[65] = 0
Norm_ij = norm([2.26901428 4.61818883]-[3 3]) = 1.7756337472247896
Norm_ij = norm([2.26901428 4.61818883]-[6 2]) = 4.557978408999554
Norm_ij = norm([2.26901428 4.61818883]-[8 5]) = 5.74369019784253
distance: [1.7756337472247896, 4.557978408999554, 5.74369019784253]
idx[66] = 0
Norm_ij = norm([-0.24512713  5.74019237]-[3 3]) = 4.247293759501029
Norm_ij = norm([-0.24512713  5.74019237]-[6 2]) = 7.2794678224608145
Norm_ij = norm([-0.24512713  5.74019237]-[8 5]) = 8.278285214376334
distance: [4.247293759501029, 7.2794678224608145, 8.278285214376334]
idx[67] = 0
Norm_ij = norm([2.12857843 5.01149793]-[3 3]) = 2.192144950833523
Norm_ij = norm([2.12857843 5.01149793]-[6 2]) = 4.904796100747739
Norm_ij = norm([2.12857843 5.01149793]-[8 5]) = 5.871432831869748
distance: [2.192144950833523, 4.904796100747739, 5.871432831869748]
idx[68] = 0
Norm_ij = norm([1.84419981 5.03153948]-[3 3]) = 2.33731186079507
Norm_ij = norm([1.84419981 5.03153948]-[6 2]) = 5.144016604557243
Norm_ij = norm([1.84419981 5.03153948]-[8 5]) = 6.15588098531526
distance: [2.33731186079507, 5.144016604557243, 6.15588098531526]
idx[69] = 0
Norm_ij = norm([2.32558253 4.74867962]-[3 3]) = 1.8742249947268799
Norm_ij = norm([2.32558253 4.74867962]-[6 2]) = 4.588745294230578
Norm_ij = norm([2.32558253 4.74867962]-[8 5]) = 5.679980240452186
distance: [1.8742249947268799, 4.588745294230578, 5.679980240452186]
```

```
idx[70] = 0
Norm_ij = norm([1.52334113 4.87916159]-[3 3]) = 2.3899309004228964
Norm_ij = norm([1.52334113 4.87916159]-[6 2]) = 5.322597684304343
Norm_ij = norm([1.52334113 4.87916159]-[8 5]) = 6.477786043486638
distance: [2.3899309004228964, 5.322597684304343, 6.477786043486638]
idx[71] = 0
Norm_ij = norm([1.02285128 5.0105065 ]-[3 3]) = 2.819796703510045
Norm_ij = norm([1.02285128 5.0105065 ]-[6 2]) = 5.816799700109842
Norm_ij = norm([1.02285128 5.0105065 ]-[8 5]) = 6.977156625939049
distance: [2.819796703510045, 5.816799700109842, 6.977156625939049]
idx[72] = 0
Norm_ij = norm([1.85382737 5.00752482]-[3 3]) = 2.3116806891188144
Norm_ij = norm([1.85382737 5.00752482]-[6 2]) = 5.122104354876449
Norm_ij = norm([1.85382737 5.00752482]-[8 5]) = 6.146177234285122
distance: [2.3116806891188144, 5.122104354876449, 6.146177234285122]
idx[73] = 0
Norm_ij = norm([2.20321658 4.94516379]-[3 3]) = 2.1020290131735333
Norm_ij = norm([2.20321658 4.94516379]-[6 2]) = 4.805159107201126
Norm_ij = norm([2.20321658 4.94516379]-[8 5]) = 5.797042779500406
distance: [2.1020290131735333, 4.805159107201126, 5.797042779500406]
idx[74] = 0
Norm_ij = norm([1.20099981 4.57829763]-[3 3]) = 2.3932039365470814
Norm_ij = norm([1.20099981 4.57829763]-[6 2]) = 5.447753800245001
Norm_ij = norm([1.20099981 4.57829763]-[8 5]) = 6.812065505223477
distance: [2.3932039365470814, 5.447753800245001, 6.812065505223477]
idx[75] = 0
Norm_ij = norm([1.02062703 4.62991119]-[3 3]) = 2.564084213840488
Norm_ij = norm([1.02062703 4.62991119]-[6 2]) = 5.63121550587325
Norm_ij = norm([1.02062703 4.62991119]-[8 5]) = 6.989178264808087
distance: [2.564084213840488, 5.63121550587325, 6.989178264808087]
idx[76] = 0
Norm_ij = norm([1.60493227 5.13663139]-[3 3]) = 2.5517460065732136
Norm_ij = norm([1.60493227 5.13663139]-[6 2]) = 5.399544135102931
Norm_ij = norm([1.60493227 5.13663139]-[8 5]) = 6.396527140414466
distance: [2.5517460065732136, 5.399544135102931, 6.396527140414466]
idx[77] = 0
Norm_ij = norm([0.47647355 5.13535977]-[3 3]) = 3.305744560116847
Norm_ij = norm([0.47647355 5.13535977]-[6 2]) = 6.351364053061753
Norm_ij = norm([0.47647355 5.13535977]-[8 5]) = 7.524744015049021
distance: [3.305744560116847, 6.351364053061753, 7.524744015049021]
idx[78] = 0
Norm_ij = norm([0.3639172  4.73332823]-[3 3]) = 3.154894490409161
Norm_ij = norm([0.3639172  4.73332823]-[6 2]) = 6.2639055287309136
Norm_ij = norm([0.3639172  4.73332823]-[8 5]) = 7.640737811297011
distance: [3.154894490409161, 6.2639055287309136, 7.640737811297011]
idx[79] = 0
Norm_ij = norm([0.31319845 5.54694644]-[3 3]) = 3.7021397480990985
Norm_ij = norm([0.31319845 5.54694644]-[6 2]) = 6.702278783018047
Norm_ij = norm([0.31319845 5.54694644]-[8 5]) = 7.706235687527162
distance: [3.7021397480990985, 6.702278783018047, 7.706235687527162]
idx[80] = 0
Norm_ij = norm([2.28664839 5.0076699 ]-[3 3]) = 2.1306358109354586
Norm_ij = norm([2.28664839 5.0076699 ]-[6 2]) = 4.778604233396704
Norm_ij = norm([2.28664839 5.0076699 ]-[8 5]) = 5.713356757222405
distance: [2.1306358109354586, 4.778604233396704, 5.713356757222405]
idx[81] = 0
Norm_ij = norm([2.15460139 5.46282959]-[3 3]) = 2.6038871718349332
Norm_ij = norm([2.15460139 5.46282959]-[6 2]) = 5.174773352270624
Norm_ij = norm([2.15460139 5.46282959]-[8 5]) = 5.863693046544135
distance: [2.6038871718349332, 5.174773352270624, 5.863693046544135]
idx[82] = 0
Norm_ij = norm([2.05288518 4.77958559]-[3 3]) = 2.015924442017656
Norm_ij = norm([2.05288518 4.77958559]-[6 2]) = 4.827609291774269
Norm_ij = norm([2.05288518 4.77958559]-[8 5]) = 5.951197966509495
distance: [2.015924442017656, 4.827609291774269, 5.951197966509495]
idx[83] = 0
Norm_ij = norm([4.88804332 5.50670795]-[3 3]) = 3.1381988979420528
Norm_ij = norm([4.88804332 5.50670795]-[6 2]) = 3.678783539175545
Norm_ij = norm([4.88804332 5.50670795]-[8 5]) = 3.1529394774425596
distance: [3.1381988979420528, 3.678783539175545, 3.1529394774425596]
idx[84] = 0
Norm_ij = norm([2.40304747 5.08147326]-[3 3]) = 2.16538293326819
Norm_ij = norm([2.40304747 5.08147326]-[6 2]) = 4.7364063340033
Norm_ij = norm([2.40304747 5.08147326]-[8 5]) = 5.597545493264205
distance: [2.16538293326819, 4.7364063340033, 5.597545493264205]
idx[85] = 0
Norm_ij = norm([2.56869453 5.20687886]-[3 3]) = 2.2486304079075428
Norm_ij = norm([2.56869453 5.20687886]-[6 2]) = 4.696586979292063
Norm_ij = norm([2.56869453 5.20687886]-[8 5]) = 5.435244058566368
distance: [2.2486304079075428, 4.696586979292063, 5.435244058566368]
idx[86] = 0
Norm_ij = norm([1.82975993 4.59657288]-[3 3]) = 1.9795218620372708
Norm_ij = norm([1.82975993 4.59657288]-[6 2]) = 4.912544454287285
Norm_ij = norm([1.82975993 4.59657288]-[8 5]) = 6.1834145920134915
distance: [1.9795218620372708, 4.912544454287285, 6.1834145920134915]
idx[87] = 0
Norm_ij = norm([0.54845223 5.0267298 ]-[3 3]) = 3.1808363912274236
Norm_ij = norm([0.54845223 5.0267298 ]-[6 2]) = 6.235420303317849
Norm_ij = norm([0.54845223 5.0267298 ]-[8 5]) = 7.451595709699586
distance: [3.1808363912274236, 6.235420303317849, 7.451595709699586]
idx[88] = 0
Norm_ij = norm([3.17109619 5.5946452 ]-[3 3]) = 2.6002802915892262
Norm_ij = norm([3.17109619 5.5946452 ]-[6 2]) = 4.574294575733014
Norm_ij = norm([3.17109619 5.5946452 ]-[8 5]) = 4.865379219225264
distance: [2.6002802915892262, 4.574294575733014, 4.865379219225264]
idx[89] = 0
```

```
Norm_ij = norm([3.04202069 5.00758373]-[3 3]) = 2.008023449307303
Norm_ij = norm([3.04202069 5.00758373]-[6 2]) = 4.2184359031800645
Norm_ij = norm([3.04202069 5.00758373]-[8 5]) = 4.957985105976522
distance: [2.008023449307303, 4.2184359031800645, 4.957985105976522]
idx[90] = 0
Norm_ij = norm([2.40427775 5.0258707 ]-[3 3]) = 2.1116432176793194
Norm_ij = norm([2.40427775 5.0258707 ]-[6 2]) = 4.699479968352952
Norm_ij = norm([2.40427775 5.0258707 ]-[8 5]) = 5.595782053817438
distance: [2.1116432176793194, 4.699479968352952, 5.595782053817438]
idx[91] = 0
Norm_ij = norm([0.17783466 5.29765032]-[3 3]) = 3.6392051554357896
Norm_ij = norm([0.17783466 5.29765032]-[6 2]) = 6.691196216811979
Norm_ij = norm([0.17783466 5.29765032]-[8 5]) = 7.827826406354568
distance: [3.6392051554357896, 6.691196216811979, 7.827826406354568]
idx[92] = 0
Norm_ij = norm([2.61428678 5.22287414]-[3 3]) = 2.2560904535354465
Norm_ij = norm([2.61428678 5.22287414]-[6 2]) = 4.674395331210469
Norm_ij = norm([2.61428678 5.22287414]-[8 5]) = 5.390322784902636
distance: [2.2560904535354465, 4.674395331210469, 5.390322784902636]
idx[93] = 0
Norm_ij = norm([2.30097798 4.97235844]-[3 3]) = 2.0925653147972945
Norm_ij = norm([2.30097798 4.97235844]-[6 2]) = 4.74527961058315
Norm_ij = norm([2.30097798 4.97235844]-[8 5]) = 5.699089051233402
distance: [2.0925653147972945, 4.74527961058315, 5.699089051233402]
idx[94] = 0
Norm_ij = norm([3.90779317 5.09464676]-[3 3]) = 2.2829002371774134
Norm_ij = norm([3.90779317 5.09464676]-[6 2]) = 3.735527804596324
Norm_ij = norm([3.90779317 5.09464676]-[8 5]) = 4.093301201914407
distance: [2.2829002371774134, 3.735527804596324, 4.093301201914407]
idx[95] = 0
Norm_ij = norm([2.05670542 5.23391326]-[3 3]) = 2.4249068205782436
Norm_ij = norm([2.05670542 5.23391326]-[6 2]) = 5.099781079052509
Norm_ij = norm([2.05670542 5.23391326]-[8 5]) = 5.947895915952168
distance: [2.4249068205782436, 5.099781079052509, 5.947895915952168]
idx[96] = 0
Norm_ij = norm([1.38133497 5.00194962]-[3 3]) = 2.5744666974971584
Norm_ij = norm([1.38133497 5.00194962]-[6 2]) = 5.508517786509164
Norm_ij = norm([1.38133497 5.00194962]-[8 5]) = 6.618665317075323
distance: [2.5744666974971584, 5.508517786509164, 6.618665317075323]
idx[97] = 0
Norm_ij = norm([1.16074178 4.67727927]-[3 3]) = 2.4892039992233723
Norm_ij = norm([1.16074178 4.67727927]-[6 2]) = 5.530483199105749
Norm_ij = norm([1.16074178 4.67727927]-[8 5]) = 6.846868020536959
distance: [2.4892039992233723, 5.530483199105749, 6.846868020536959]
idx[98] = 0
Norm_ij = norm([1.72818199 5.36028437]-[3 3]) = 2.6811309865945723
Norm_ij = norm([1.72818199 5.36028437]-[6 2]) = 5.435065790759416
Norm_ij = norm([1.72818199 5.36028437]-[8 5]) = 6.2821577444718075
```

```
distance: [2.6811309865945723, 5.435065790759416, 6.2821577444718075]
idx[99] = 0
Norm_ij = norm([3.20360621 0.7222149 ]-[3 3]) = 2.2868669469858913
Norm_ij = norm([3.20360621 0.7222149 ]-[6 2]) = 3.0745004481047653
Norm_ij = norm([3.20360621 0.7222149 ]-[8 5]) = 6.426884063194449
distance: [2.2868669469858913, 3.0745004481047653, 6.426884063194449]
idx[100] = 0
Norm_ij = norm([3.06192918 1.5719211 ]-[3 3]) = 1.4294210613468505
Norm_ij = norm([3.06192918 1.5719211 ]-[6 2]) = 2.969092734182235
Norm_ij = norm([3.06192918 1.5719211 ]-[8 5]) = 6.011344968644903
distance: [1.4294210613468505, 2.969092734182235, 6.011344968644903]
idx[101] = 0
Norm_ij = norm([4.01714917 1.16070647]-[3 3]) = 2.1018071106432594
Norm_ij = norm([4.01714917 1.16070647]-[6 2]) = 2.1531630335951046
Norm_ij = norm([4.01714917 1.16070647]-[8 5]) = 5.532022737141138
distance: [2.1018071106432594, 2.1531630335951046, 5.532022737141138]
idx[102] = 0
Norm_ij = norm([1.40260822 1.08726536]-[3 3]) = 2.492030158578039
Norm_ij = norm([1.40260822 1.08726536]-[6 2]) = 4.687120193798859
Norm_ij = norm([1.40260822 1.08726536]-[8 5]) = 7.670402250548817
distance: [2.492030158578039, 4.687120193798859, 7.670402250548817]
idx[103] = 0
Norm_ij = norm([4.08164951 0.87200343]-[3 3]) = 2.387118569412935
Norm_ij = norm([4.08164951 0.87200343]-[6 2]) = 2.2254089234868233
Norm_ij = norm([4.08164951 0.87200343]-[8 5]) = 5.691557455932148
distance: [2.387118569412935, 2.2254089234868233, 5.691557455932148]
idx[104] = 1
Norm_ij = norm([3.15273081 0.98155871]-[3 3]) = 2.0242114353769525
Norm_ij = norm([3.15273081 0.98155871]-[6 2]) = 3.0239319613719147
Norm_ij = norm([3.15273081 0.98155871]-[8 5]) = 6.296339333625429
distance: [2.0242114353769525, 3.0239319613719147, 6.296339333625429]
idx[105] = 0
Norm_ij = norm([3.45186351 0.42784083]-[3 3]) = 2.6115480856766733
Norm_ij = norm([3.45186351 0.42784083]-[6 2]) = 2.9941082176617484
Norm_ij = norm([3.45186351 0.42784083]-[8 5]) = 6.449045276385476
distance: [2.6115480856766733, 2.9941082176617484, 6.449045276385476]
idx[106] = 0
Norm_ij = norm([3.85384314 0.7920479 ]-[3 3]) = 2.36729815712204
Norm_ij = norm([3.85384314 0.7920479 ]-[6 2]) = 2.4627499966207305
Norm_ij = norm([3.85384314 0.7920479 ]-[8 5]) = 5.907408702246601
distance: [2.36729815712204, 2.4627499966207305, 5.907408702246601]
idx[107] = 0
Norm_ij = norm([1.57449255 1.34811126]-[3 3]) = 2.181927564693292
Norm_ij = norm([1.57449255 1.34811126]-[6 2]) = 4.4732622446346975
Norm_ij = norm([1.57449255 1.34811126]-[8 5]) = 7.390767033771075
distance: [2.181927564693292, 4.4732622446346975, 7.390767033771075]
idx[108] = 0
Norm_ij = norm([4.72372078 0.62044136]-[3 3]) = 2.938283963048568
Norm_ij = norm([4.72372078 0.62044136]-[6 2]) = 1.8793804029764365
Norm_ij = norm([4.72372078 0.62044136]-[8 5]) = 5.469418564903575
distance: [2.938283963048568, 1.8793804029764365, 5.469418564903575]
idx[109] = 1
Norm_ij = norm([2.87961084 0.75413741]-[3 3]) = 2.249086996550985
Norm_ij = norm([2.87961084 0.75413741]-[6 2]) = 3.3599110250271824
Norm_ij = norm([2.87961084 0.75413741]-[8 5]) = 6.651746707969124
distance: [2.249086996550985, 3.3599110250271824, 6.651746707969124]
idx[110] = 0
Norm_ij = norm([0.96791348 1.16166819]-[3 3]) = 2.740226170260796
Norm_ij = norm([0.96791348 1.16166819]-[6 2]) = 5.101440482188394
Norm_ij = norm([0.96791348 1.16166819]-[8 5]) = 8.011431327658624
distance: [2.740226170260796, 5.101440482188394, 8.011431327658624]
idx[111] = 0
Norm_ij = norm([1.53178107 1.10054852]-[3 3]) = 2.4007462868428244
Norm_ij = norm([1.53178107 1.10054852]-[6 2]) = 4.557849642030869
Norm_ij = norm([1.53178107 1.10054852]-[8 5]) = 7.552719902924839
distance: [2.4007462868428244, 4.557849642030869, 7.552719902924839]
idx[112] = 0
Norm_ij = norm([4.13835915 1.24780979]-[3 3]) = 2.089505223725696
Norm_ij = norm([4.13835915 1.24780979]-[6 2]) = 2.0078587483700505
Norm_ij = norm([4.13835915 1.24780979]-[8 5]) = 5.384347814284158
distance: [2.089505223725696, 2.0078587483700505, 5.384347814284158]
idx[113] = 1
Norm_ij = norm([3.16109021 1.29422893]-[3 3]) = 1.7133607309249952
Norm_ij = norm([3.16109021 1.29422893]-[6 2]) = 2.9253241818754514
Norm_ij = norm([3.16109021 1.29422893]-[8 5]) = 6.094898450241293
distance: [1.7133607309249952, 2.9253241818754514, 6.094898450241293]
idx[114] = 0
Norm_ij = norm([2.95177039 0.89583143]-[3 3]) = 2.1047212299910267
Norm_ij = norm([2.95177039 0.89583143]-[6 2]) = 3.2420505835670768
Norm_ij = norm([2.95177039 0.89583143]-[8 5]) = 6.506060392840817
distance: [2.1047212299910267, 3.2420505835670768, 6.506060392840817]
idx[115] = 0
Norm_ij = norm([3.27844295 1.75043926]-[3 3]) = 1.280207997994104
Norm_ij = norm([3.27844295 1.75043926]-[6 2]) = 2.73297517614649
Norm_ij = norm([3.27844295 1.75043926]-[8 5]) = 5.731731494567589
distance: [1.280207997994104, 2.73297517614649, 5.731731494567589]
idx[116] = 0
Norm_ij = norm([2.1270185  0.95672042]-[3 3]) = 2.221955922678612
Norm_ij = norm([2.1270185  0.95672042]-[6 2]) = 4.011037020070031
Norm_ij = norm([2.1270185  0.95672042]-[8 5]) = 7.130218890705635
distance: [2.221955922678612, 4.011037020070031, 7.130218890705635]
idx[117] = 0
Norm_ij = norm([3.32648885 1.28019066]-[3 3]) = 1.7505253840842272
Norm_ij = norm([3.32648885 1.28019066]-[6 2]) = 2.7687158321888643
Norm_ij = norm([3.32648885 1.28019066]-[8 5]) = 5.973163984286535
distance: [1.7505253840842272, 2.7687158321888643, 5.973163984286535]
```

```
idx[118] = 0
Norm_ij = norm([2.54371489 0.95732716]-[3 3]) = 2.0930141979460966
Norm_ij = norm([2.54371489 0.95732716]-[6 2]) = 3.6101348151272847
Norm_ij = norm([2.54371489 0.95732716]-[8 5]) = 6.7907474442163664
distance: [2.0930141979460966, 3.6101348151272847, 6.7907474442163664]
idx[119] = 0
Norm_ij = norm([3.233947   1.08202324]-[3 3]) = 1.9321920323770188
Norm_ij = norm([3.233947   1.08202324]-[6 2]) = 2.9144005440173784
Norm_ij = norm([3.233947   1.08202324]-[8 5]) = 6.169749029768567
distance: [1.9321920323770188, 2.9144005440173784, 6.169749029768567]
idx[120] = 0
Norm_ij = norm([4.43152976 0.54041   ]-[3 3]) = 2.8458496823746384
Norm_ij = norm([4.43152976 0.54041   ]-[6 2]) = 2.142545645842187
Norm_ij = norm([4.43152976 0.54041   ]-[8 5]) = 5.711560450129171
distance: [2.8458496823746384, 2.142545645842187, 5.711560450129171]
idx[121] = 1
Norm_ij = norm([3.56478625 1.11764714]-[3 3]) = 1.965257182185948
Norm_ij = norm([3.56478625 1.11764714]-[6 2]) = 2.590137555049203
Norm_ij = norm([3.56478625 1.11764714]-[8 5]) = 5.8943858613554685
distance: [1.965257182185948, 2.590137555049203, 5.8943858613554685]
idx[122] = 0
Norm_ij = norm([4.25588482 0.90643957]-[3 3]) = 2.4413606755829482
Norm_ij = norm([4.25588482 0.90643957]-[6 2]) = 2.058594706793277
Norm_ij = norm([4.25588482 0.90643957]-[8 5]) = 5.547579242288998
distance: [2.4413606755829482, 2.058594706793277, 5.547579242288998]
idx[123] = 1
Norm_ij = norm([4.05386581 0.53291862]-[3 3]) = 2.682745549462226
Norm_ij = norm([4.05386581 0.53291862]-[6 2]) = 2.437163524439984
Norm_ij = norm([4.05386581 0.53291862]-[8 5]) = 5.960435476138678
distance: [2.682745549462226, 2.437163524439984, 5.960435476138678]
idx[124] = 1
Norm_ij = norm([3.08970176 1.08814448]-[3 3]) = 1.9139587054220144
Norm_ij = norm([3.08970176 1.08814448]-[6 2]) = 3.0498059494347247
Norm_ij = norm([3.08970176 1.08814448]-[8 5]) = 6.278028543569551
distance: [1.9139587054220144, 3.0498059494347247, 6.278028543569551]
idx[125] = 0
Norm_ij = norm([2.84734459 0.26759253]-[3 3]) = 2.736668457219526
Norm_ij = norm([2.84734459 0.26759253]-[6 2]) = 3.5972867203780714
Norm_ij = norm([2.84734459 0.26759253]-[8 5]) = 6.996108788725925
distance: [2.736668457219526, 3.5972867203780714, 6.996108788725925]
idx[126] = 0
Norm_ij = norm([3.63586049 1.12160194]-[3 3]) = 1.9831030778870882
Norm_ij = norm([3.63586049 1.12160194]-[6 2]) = 2.522050511578826
Norm_ij = norm([3.63586049 1.12160194]-[8 5]) = 5.838465994959429
distance: [1.9831030778870882, 2.522050511578826, 5.838465994959429]
idx[127] = 0
Norm_ij = norm([1.95538864 1.32156857]-[3 3]) = 1.976953456000709
Norm_ij = norm([1.95538864 1.32156857]-[6 2]) = 4.101115736230716
Norm_ij = norm([1.95538864 1.32156857]-[8 5]) = 7.075887528639233
distance: [1.976953456000709, 4.101115736230716, 7.075887528639233]
idx[128] = 0
Norm_ij = norm([2.88384005 0.80454506]-[3 3]) = 2.198525757492376
Norm_ij = norm([2.88384005 0.80454506]-[6 2]) = 3.337598740440629
Norm_ij = norm([2.88384005 0.80454506]-[8 5]) = 6.6164140418344175
distance: [2.198525757492376, 3.337598740440629, 6.6164140418344175]
idx[129] = 0
Norm_ij = norm([3.48444387 1.13551448]-[3 3]) = 1.9263935504156446
Norm_ij = norm([3.48444387 1.13551448]-[6 2]) = 2.6599544806660025
Norm_ij = norm([3.48444387 1.13551448]-[8 5]) = 5.9434413813400155
distance: [1.9263935504156446, 2.6599544806660025, 5.9434413813400155]
idx[130] = 0
Norm_ij = norm([3.49798412 1.10046402]-[3 3]) = 1.9637273547769825
Norm_ij = norm([3.49798412 1.10046402]-[6 2]) = 2.658805825439417
Norm_ij = norm([3.49798412 1.10046402]-[8 5]) = 5.956049680126054
distance: [1.9637273547769825, 2.658805825439417, 5.956049680126054]
idx[131] = 0
Norm_ij = norm([2.45575934 0.78904654]-[3 3]) = 2.2769525887496793
Norm_ij = norm([2.45575934 0.78904654]-[6 2]) = 3.745403868325655
Norm_ij = norm([2.45575934 0.78904654]-[8 5]) = 6.962092611310221
distance: [2.2769525887496793, 3.745403868325655, 6.962092611310221]
idx[132] = 0
Norm_ij = norm([3.2038001  1.02728075]-[3 3]) = 1.9832185286877577
Norm_ij = norm([3.2038001  1.02728075]-[6 2]) = 2.9605601925350413
Norm_ij = norm([3.2038001  1.02728075]-[8 5]) = 6.227843269897605
distance: [1.9832185286877577, 2.9605601925350413, 6.227843269897605]
idx[133] = 0
Norm_ij = norm([3.00677254 0.62519128]-[3 3]) = 2.3748183754812136
Norm_ij = norm([3.00677254 0.62519128]-[6 2]) = 3.29385938655578
Norm_ij = norm([3.00677254 0.62519128]-[8 5]) = 6.63861972262944
distance: [2.3748183754812136, 3.29385938655578, 6.63861972262944]
idx[134] = 0
Norm_ij = norm([1.96547974 1.2173076 ]-[3 3]) = 2.0611221091294447
Norm_ij = norm([1.96547974 1.2173076 ]-[6 2]) = 4.1097397889582625
Norm_ij = norm([1.96547974 1.2173076 ]-[8 5]) = 7.122092147181045
distance: [2.0611221091294447, 4.1097397889582625, 7.122092147181045]
idx[135] = 0
Norm_ij = norm([2.17989333 1.30879831]-[3 3]) = 1.87955795862197
Norm_ij = norm([2.17989333 1.30879831]-[6 2]) = 3.882135332281059
Norm_ij = norm([2.17989333 1.30879831]-[8 5]) = 6.891923648715924
distance: [1.87955795862197, 3.882135332281059, 6.891923648715924]
idx[136] = 0
Norm_ij = norm([2.61207029 0.99076856]-[3 3]) = 2.046338302266225
Norm_ij = norm([2.61207029 0.99076856]-[6 2]) = 3.535055275802883
Norm_ij = norm([2.61207029 0.99076856]-[8 5]) = 6.715930559926856
distance: [2.046338302266225, 3.535055275802883, 6.715930559926856]
idx[137] = 0
```

```
Norm_ij = norm([3.95549912 0.83269299]-[3 3]) = 2.3685857038158074
Norm_ij = norm([3.95549912 0.83269299]-[6 2]) = 2.35427048646898
Norm_ij = norm([3.95549912 0.83269299]-[8 5]) = 5.807274328256759
distance: [2.3685857038158074, 2.35427048646898, 5.807274328256759]
idx[138] = 1
Norm_ij = norm([3.64846482 1.62849697]-[3 3]) = 1.5170784975670875
Norm_ij = norm([3.64846482 1.62849697]-[6 2]) = 2.3806999426875914
Norm_ij = norm([3.64846482 1.62849697]-[8 5]) = 5.504806182334712
distance: [1.5170784975670875, 2.3806999426875914, 5.504806182334712]
idx[139] = 0
Norm_ij = norm([4.18450011 0.45356203]-[3 3]) = 2.8084491861707677
Norm_ij = norm([4.18450011 0.45356203]-[6 2]) = 2.3848501534784576
Norm_ij = norm([4.18450011 0.45356203]-[8 5]) = 5.935329615213928
distance: [2.8084491861707677, 2.3848501534784576, 5.935329615213928]
idx[140] = 1
Norm_ij = norm([3.7875723  1.45442904]-[3 3]) = 1.7346641487358816
Norm_ij = norm([3.7875723  1.45442904]-[6 2]) = 2.2787022594919994
Norm_ij = norm([3.7875723  1.45442904]-[8 5]) = 5.505962269508427
distance: [1.7346641487358816, 2.2787022594919994, 5.505962269508427]
idx[141] = 0
Norm_ij = norm([3.30063655 1.28107588]-[3 3]) = 1.745016463586638
Norm_ij = norm([3.30063655 1.28107588]-[6 2]) = 2.793459311171795
Norm_ij = norm([3.30063655 1.28107588]-[8 5]) = 5.992863541833187
distance: [1.745016463586638, 2.793459311171795, 5.992863541833187]
idx[142] = 0
Norm_ij = norm([3.02836363 1.35635189]-[3 3]) = 1.6438928205881478
Norm_ij = norm([3.02836363 1.35635189]-[6 2]) = 3.0405436341843424
Norm_ij = norm([3.02836363 1.35635189]-[8 5]) = 6.1638737594862905
distance: [1.6438928205881478, 3.0405436341843424, 6.1638737594862905]
idx[143] = 0
Norm_ij = norm([3.18412176 1.41410799]-[3 3]) = 1.596544484979842
Norm_ij = norm([3.18412176 1.41410799]-[6 2]) = 2.8761849267292625
Norm_ij = norm([3.18412176 1.41410799]-[8 5]) = 6.004273875794164
distance: [1.596544484979842, 2.8761849267292625, 6.004273875794164]
idx[144] = 0
Norm_ij = norm([4.16911897 0.20581038]-[3 3]) = 3.0289164405448523
Norm_ij = norm([4.16911897 0.20581038]-[6 2]) = 2.563443340783047
Norm_ij = norm([4.16911897 0.20581038]-[8 5]) = 6.136766545310445
distance: [3.0289164405448523, 2.563443340783047, 6.136766545310445]
idx[145] = 1
Norm_ij = norm([3.24024211 1.14876237]-[3 3]) = 1.8667611053371014
Norm_ij = norm([3.24024211 1.14876237]-[6 2]) = 2.8880562847378615
Norm_ij = norm([3.24024211 1.14876237]-[8 5]) = 6.122689476296379
distance: [1.8667611053371014, 2.8880562847378615, 6.122689476296379]
idx[146] = 0
Norm_ij = norm([3.91596068 1.01225774]-[3 3]) = 2.1886304560928043
Norm_ij = norm([3.91596068 1.01225774]-[6 2]) = 2.306264225474707
Norm_ij = norm([3.91596068 1.01225774]-[8 5]) = 5.708017655341244
```

```
distance: [2.1886304560928043, 2.306264225474707, 5.708017655341244]
idx[147] = 0
Norm_ij = norm([2.96979716 1.01210306]-[3 3]) = 1.988126368160162
Norm_ij = norm([2.96979716 1.01210306]-[6 2]) = 3.1871726662339404
Norm_ij = norm([2.96979716 1.01210306]-[8 5]) = 6.419210434104649
distance: [1.988126368160162, 3.1871726662339404, 6.419210434104649]
idx[148] = 0
Norm_ij = norm([1.12993856 0.77085284]-[3 3]) = 2.9096781342561795
Norm_ij = norm([1.12993856 0.77085284]-[6 2]) = 5.022778234214831
Norm_ij = norm([1.12993856 0.77085284]-[8 5]) = 8.067430191817705
distance: [2.9096781342561795, 5.022778234214831, 8.067430191817705]
idx[149] = 0
Norm_ij = norm([2.71730799 0.48697555]-[3 3]) = 2.528874587654738
Norm_ij = norm([2.71730799 0.48697555]-[6 2]) = 3.614596776194319
Norm_ij = norm([2.71730799 0.48697555]-[8 5]) = 6.947965503085992
distance: [2.528874587654738, 3.614596776194319, 6.947965503085992]
idx[150] = 0
Norm_ij = norm([3.1189017  0.69438336]-[3 3]) = 2.308680515917855
Norm_ij = norm([3.1189017  0.69438336]-[6 2]) = 3.1631253943276123
Norm_ij = norm([3.1189017  0.69438336]-[8 5]) = 6.508721483019217
distance: [2.308680515917855, 3.1631253943276123, 6.508721483019217]
idx[151] = 0
Norm_ij = norm([2.4051802  1.11778123]-[3 3]) = 1.9739701375733534
Norm_ij = norm([2.4051802  1.11778123]-[6 2]) = 3.7014915042772403
Norm_ij = norm([2.4051802  1.11778123]-[8 5]) = 6.8098187342901255
distance: [1.9739701375733534, 3.7014915042772403, 6.8098187342901255]
idx[152] = 0
Norm_ij = norm([2.95818429 1.01887096]-[3 3]) = 1.9815702962673105
Norm_ij = norm([2.95818429 1.01887096]-[6 2]) = 3.1961315692687866
Norm_ij = norm([2.95818429 1.01887096]-[8 5]) = 6.424118157667686
distance: [1.9815702962673105, 3.1961315692687866, 6.424118157667686]
idx[153] = 0
Norm_ij = norm([1.65456309 1.18631175]-[3 3]) = 2.258243912628122
Norm_ij = norm([1.65456309 1.18631175]-[6 2]) = 4.420962627488467
Norm_ij = norm([1.65456309 1.18631175]-[8 5]) = 7.403295734179963
distance: [2.258243912628122, 4.420962627488467, 7.403295734179963]
idx[154] = 0
Norm_ij = norm([2.39775807 1.24721387]-[3 3]) = 1.8533630377103985
Norm_ij = norm([2.39775807 1.24721387]-[6 2]) = 3.6800589516653015
Norm_ij = norm([2.39775807 1.24721387]-[8 5]) = 6.743034804434937
distance: [1.8533630377103985, 3.6800589516653015, 6.743034804434937]
idx[155] = 0
Norm_ij = norm([2.28409305 0.64865469]-[3 3]) = 2.457915280461451
Norm_ij = norm([2.28409305 0.64865469]-[6 2]) = 3.9539978004782412
Norm_ij = norm([2.28409305 0.64865469]-[8 5]) = 7.1837175795999455
distance: [2.457915280461451, 3.9539978004782412, 7.1837175795999455]
idx[156] = 0
Norm_ij = norm([2.79588724 0.99526664]-[3 3]) = 2.015097485498646
Norm_ij = norm([2.79588724 0.99526664]-[6 2]) = 3.3579499237940156
Norm_ij = norm([2.79588724 0.99526664]-[8 5]) = 6.566633757039816
distance: [2.015097485498646, 3.3579499237940156, 6.566633757039816]
idx[157] = 0
Norm_ij = norm([3.41156277 1.1596363 ]-[3 3]) = 1.885821431616669
Norm_ij = norm([3.41156277 1.1596363 ]-[6 2]) = 2.7214368340513304
Norm_ij = norm([3.41156277 1.1596363 ]-[8 5]) = 5.983489748756229
distance: [1.885821431616669, 2.7214368340513304, 5.983489748756229]
idx[158] = 0
Norm_ij = norm([3.50663521 0.73878104]-[3 3]) = 2.3172808213302827
Norm_ij = norm([3.50663521 0.73878104]-[6 2]) = 2.794197776952592
Norm_ij = norm([3.50663521 0.73878104]-[8 5]) = 6.192601561582774
distance: [2.3172808213302827, 2.794197776952592, 6.192601561582774]
idx[159] = 0
Norm_ij = norm([3.93616029 1.46202934]-[3 3]) = 1.8004859996295564
Norm_ij = norm([3.93616029 1.46202934]-[6 2]) = 2.132802565784627
Norm_ij = norm([3.93616029 1.46202934]-[8 5]) = 5.388137858859404
distance: [1.8004859996295564, 2.132802565784627, 5.388137858859404]
idx[160] = 0
Norm_ij = norm([3.90206657 1.27778751]-[3 3]) = 1.9441553364198074
Norm_ij = norm([3.90206657 1.27778751]-[6 2]) = 2.2187644199261642
Norm_ij = norm([3.90206657 1.27778751]-[8 5]) = 5.53605673912185
distance: [1.9441553364198074, 2.2187644199261642, 5.53605673912185]
idx[161] = 0
Norm_ij = norm([2.61036396 0.88027602]-[3 3]) = 2.1552368778237865
Norm_ij = norm([2.61036396 0.88027602]-[6 2]) = 3.569791908975396
Norm_ij = norm([2.61036396 0.88027602]-[8 5]) = 6.783826524749029
distance: [2.1552368778237865, 3.569791908975396, 6.783826524749029]
idx[162] = 0
Norm_ij = norm([4.37271861 1.02914092]-[3 3]) = 2.40179972062982
Norm_ij = norm([4.37271861 1.02914092]-[6 2]) = 1.8948910457500083
Norm_ij = norm([4.37271861 1.02914092]-[8 5]) = 5.378186694098119
distance: [2.40179972062982, 1.8948910457500083, 5.378186694098119]
idx[163] = 1
Norm_ij = norm([3.08349136 1.19632644]-[3 3]) = 1.8056049133301757
Norm_ij = norm([3.08349136 1.19632644]-[6 2]) = 3.025213019448012
Norm_ij = norm([3.08349136 1.19632644]-[8 5]) = 6.216107278226315
distance: [1.8056049133301757, 3.025213019448012, 6.216107278226315]
idx[164] = 0
Norm_ij = norm([2.1159935 0.7930365]-[3 3]) = 2.377426208986817
Norm_ij = norm([2.1159935 0.7930365]-[6 2]) = 4.067218628230811
Norm_ij = norm([2.1159935 0.7930365]-[8 5]) = 7.2332616683586615
distance: [2.377426208986817, 4.067218628230811, 7.2332616683586615]
idx[165] = 0
Norm_ij = norm([2.15653404 0.40358861]-[3 3]) = 2.729979292964998
Norm_ij = norm([2.15653404 0.40358861]-[6 2]) = 4.161821702627584
Norm_ij = norm([2.15653404 0.40358861]-[8 5]) = 7.434587551704987
distance: [2.729979292964998, 4.161821702627584, 7.434587551704987]
```

```
idx[166] = 0
Norm_ij = norm([2.14491101 1.13582399]-[3 3]) = 2.050933780584805
Norm_ij = norm([2.14491101 1.13582399]-[6 2]) = 3.9507608486528407
Norm_ij = norm([2.14491101 1.13582399]-[8 5]) = 7.015263594016222
distance: [2.050933780584805, 3.9507608486528407, 7.015263594016222]
idx[167] = 0
Norm_ij = norm([1.84935524 1.02232644]-[3 3]) = 2.288050717059634
Norm_ij = norm([1.84935524 1.02232644]-[6 2]) = 4.264234692561707
Norm_ij = norm([1.84935524 1.02232644]-[8 5]) = 7.324774257841074
distance: [2.288050717059634, 4.264234692561707, 7.324774257841074]
idx[168] = 0
Norm_ij = norm([4.1590816  0.61720733]-[3 3]) = 2.6497492505223774
Norm_ij = norm([4.1590816  0.61720733]-[6 2]) = 2.302410939018421
Norm_ij = norm([4.1590816  0.61720733]-[8 5]) = 5.82765182351842
distance: [2.6497492505223774, 2.302410939018421, 5.82765182351842]
idx[169] = 1
Norm_ij = norm([2.76494499 1.43148951]-[3 3]) = 1.5860252290870074
Norm_ij = norm([2.76494499 1.43148951]-[6 2]) = 3.28462860677354
Norm_ij = norm([2.76494499 1.43148951]-[8 5]) = 6.3356189974336505
distance: [1.5860252290870074, 3.28462860677354, 6.3356189974336505]
idx[170] = 0
Norm_ij = norm([3.90561153 1.16575315]-[3 3]) = 2.0456279607406276
Norm_ij = norm([3.90561153 1.16575315]-[6 2]) = 2.254424729925599
Norm_ij = norm([3.90561153 1.16575315]-[8 5]) = 5.609408688695045
distance: [2.0456279607406276, 2.254424729925599, 5.609408688695045]
idx[171] = 0
Norm_ij = norm([2.54071672 0.98392516]-[3 3]) = 2.0677279543451164
Norm_ij = norm([2.54071672 0.98392516]-[6 2]) = 3.6054193741603946
Norm_ij = norm([2.54071672 0.98392516]-[8 5]) = 6.777361655827457
distance: [2.0677279543451164, 3.6054193741603946, 6.777361655827457]
idx[172] = 0
Norm_ij = norm([4.27783068 1.1801368 ]-[3 3]) = 2.22368013189733
Norm_ij = norm([4.27783068 1.1801368 ]-[6 2]) = 1.9073654169936167
Norm_ij = norm([4.27783068 1.1801368 ]-[8 5]) = 5.333469726913179
distance: [2.22368013189733, 1.9073654169936167, 5.333469726913179]
idx[173] = 1
Norm_ij = norm([3.31058167 1.03124461]-[3 3]) = 1.9931027973853266
Norm_ij = norm([3.31058167 1.03124461]-[6 2]) = 2.858576213036053
Norm_ij = norm([3.31058167 1.03124461]-[8 5]) = 6.143424422335909
distance: [1.9931027973853266, 2.858576213036053, 6.143424422335909]
idx[174] = 0
Norm_ij = norm([2.15520661 0.80696562]-[3 3]) = 2.3501224738931046
Norm_ij = norm([2.15520661 0.80696562]-[6 2]) = 4.025638732995598
Norm_ij = norm([2.15520661 0.80696562]-[8 5]) = 7.193270953713542
distance: [2.3501224738931046, 4.025638732995598, 7.193270953713542]
idx[175] = 0
Norm_ij = norm([3.71363659 0.45813208]-[3 3]) = 2.6401457714282937
Norm_ij = norm([3.71363659 0.45813208]-[6 2]) = 2.7576827825178785
Norm_ij = norm([3.71363659 0.45813208]-[8 5]) = 6.245116130776154
distance: [2.6401457714282937, 2.7576827825178785, 6.245116130776154]
idx[176] = 0
Norm_ij = norm([3.54010186 0.86446135]-[3 3]) = 2.202779003875376
Norm_ij = norm([3.54010186 0.86446135]-[6 2]) = 2.709344360298427
Norm_ij = norm([3.54010186 0.86446135]-[8 5]) = 6.082217629794025
distance: [2.202779003875376, 2.709344360298427, 6.082217629794025]
idx[177] = 0
Norm_ij = norm([1.60519991 1.1098053 ]-[3 3]) = 2.349106909958088
Norm_ij = norm([1.60519991 1.1098053 ]-[6 2]) = 4.4840511138718115
Norm_ij = norm([1.60519991 1.1098053 ]-[8 5]) = 7.485124108596916
distance: [2.349106909958088, 4.4840511138718115, 7.485124108596916]
idx[178] = 0
Norm_ij = norm([1.75164337 0.68853536]-[3 3]) = 2.627025513685169
Norm_ij = norm([1.75164337 0.68853536]-[6 2]) = 4.446175158716932
Norm_ij = norm([1.75164337 0.68853536]-[8 5]) = 7.591487857483219
distance: [2.627025513685169, 4.446175158716932, 7.591487857483219]
idx[179] = 0
Norm_ij = norm([3.12405123 0.67821757]-[3 3]) = 2.325094057271909
Norm_ij = norm([3.12405123 0.67821757]-[6 2]) = 3.165152468569918
Norm_ij = norm([3.12405123 0.67821757]-[8 5]) = 6.5155721041332955
distance: [2.325094057271909, 3.165152468569918, 6.5155721041332955]
idx[180] = 0
Norm_ij = norm([2.37198785 1.42789607]-[3 3]) = 1.6928998841065455
Norm_ij = norm([2.37198785 1.42789607]-[6 2]) = 3.672842910345086
Norm_ij = norm([2.37198785 1.42789607]-[8 5]) = 6.665916831239001
distance: [1.6928998841065455, 3.672842910345086, 6.665916831239001]
idx[181] = 0
Norm_ij = norm([2.53446019 1.21562081]-[3 3]) = 1.8441085640094572
Norm_ij = norm([2.53446019 1.21562081]-[6 2]) = 3.5531981178507843
Norm_ij = norm([2.53446019 1.21562081]-[8 5]) = 6.64783056468912
distance: [1.8441085640094572, 3.5531981178507843, 6.64783056468912]
idx[182] = 0
Norm_ij = norm([3.6834465  1.22834538]-[3 3]) = 1.8989100115724424
Norm_ij = norm([3.6834465  1.22834538]-[6 2]) = 2.4416942810717446
Norm_ij = norm([3.6834465  1.22834538]-[8 5]) = 5.732190915902187
distance: [1.8989100115724424, 2.4416942810717446, 5.732190915902187]
idx[183] = 0
Norm_ij = norm([3.2670134  0.32056676]-[3 3]) = 2.6927047087987677
Norm_ij = norm([3.2670134  0.32056676]-[6 2]) = 3.207758062625057
Norm_ij = norm([3.2670134  0.32056676]-[8 5]) = 6.65569362662238
distance: [2.6927047087987677, 3.207758062625057, 6.65569362662238]
idx[184] = 0
Norm_ij = norm([3.94159139 0.82577438]-[3 3]) = 2.3693567473632244
Norm_ij = norm([3.94159139 0.82577438]-[6 2]) = 2.369778851835372
Norm_ij = norm([3.94159139 0.82577438]-[8 5]) = 5.8219275125892365
distance: [2.3693567473632244, 2.369778851835372, 5.8219275125892365]
idx[185] = 0
```

```
Norm_ij = norm([3.2645514 1.3836869]-[3 3]) = 1.6378203462406749
Norm_ij = norm([3.2645514 1.3836869]-[6 2]) = 2.804018700384854
Norm_ij = norm([3.2645514 1.3836869]-[8 5]) = 5.958371746880124
distance: [1.6378203462406749, 2.804018700384854, 5.958371746880124]
idx[186] = 0
Norm_ij = norm([4.30471138 1.10725995]-[3 3]) = 2.2988555163811366
Norm_ij = norm([4.30471138 1.10725995]-[6 2]) = 1.915982330841848
Norm_ij = norm([4.30471138 1.10725995]-[8 5]) = 5.367362765561803
distance: [2.2988555163811366, 1.915982330841848, 5.367362765561803]
idx[187] = 1
Norm_ij = norm([2.68499376 0.35344943]-[3 3]) = 2.665231478752315
Norm_ij = norm([2.68499376 0.35344943]-[6 2]) = 3.7014044842980205
Norm_ij = norm([2.68499376 0.35344943]-[8 5]) = 7.059725456903312
distance: [2.665231478752315, 3.7014044842980205, 7.059725456903312]
idx[188] = 0
Norm_ij = norm([3.12635184 1.2806893 ]-[3 3]) = 1.7239472379490768
Norm_ij = norm([3.12635184 1.2806893 ]-[6 2]) = 2.962306809166157
Norm_ij = norm([3.12635184 1.2806893 ]-[8 5]) = 6.130719246933804
distance: [1.7239472379490768, 2.962306809166157, 6.130719246933804]
idx[189] = 0
Norm_ij = norm([2.94294356 1.02825076]-[3 3]) = 1.9725745849154195
Norm_ij = norm([2.94294356 1.02825076]-[6 2]) = 3.2077859462015548
Norm_ij = norm([2.94294356 1.02825076]-[8 5]) = 6.43028863066244
distance: [1.9725745849154195, 3.2077859462015548, 6.43028863066244]
idx[190] = 0
Norm_ij = norm([3.11876541 1.33285459]-[3 3]) = 1.6713704122822923
Norm_ij = norm([3.11876541 1.33285459]-[6 2]) = 2.9574644141763713
Norm_ij = norm([3.11876541 1.33285459]-[8 5]) = 6.1052769472752555
distance: [1.6713704122822923, 2.9574644141763713, 6.1052769472752555]
idx[191] = 0
Norm_ij = norm([2.02358978 0.44771614]-[3 3]) = 2.732678142572782
Norm_ij = norm([2.02358978 0.44771614]-[6 2]) = 4.268655928117128
Norm_ij = norm([2.02358978 0.44771614]-[8 5]) = 7.512707067849015
distance: [2.732678142572782, 4.268655928117128, 7.512707067849015]
idx[192] = 0
Norm_ij = norm([3.62202931 1.28643763]-[3 3]) = 1.8229691340579692
Norm_ij = norm([3.62202931 1.28643763]-[6 2]) = 2.4827234716310516
Norm_ij = norm([3.62202931 1.28643763]-[8 5]) = 5.740833806845178
distance: [1.8229691340579692, 2.4827234716310516, 5.740833806845178]
idx[193] = 0
Norm_ij = norm([2.42865879 0.86499285]-[3 3]) = 2.2101326506445553
Norm_ij = norm([2.42865879 0.86499285]-[6 2]) = 3.747361646183936
Norm_ij = norm([2.42865879 0.86499285]-[8 5]) = 6.938164533306308
distance: [2.2101326506445553, 3.747361646183936, 6.938164533306308]
idx[194] = 0
Norm_ij = norm([2.09517296 1.14010491]-[3 3]) = 2.0683137360750767
Norm_ij = norm([2.09517296 1.14010491]-[6 2]) = 3.998386394752841
Norm_ij = norm([2.09517296 1.14010491]-[8 5]) = 7.054485980422536
```

```
distance: [2.0683137360750767, 3.998386394752841, 7.054485980422536]
idx[195] = 0
Norm_ij = norm([5.29239452 0.36873298]-[3 3]) = 3.4897906484860264
Norm_ij = norm([5.29239452 0.36873298]-[6 2]) = 1.778127555330587
Norm_ij = norm([5.29239452 0.36873298]-[8 5]) = 5.364677216382843
distance: [3.4897906484860264, 1.778127555330587, 5.364677216382843]
idx[196] = 1
Norm_ij = norm([2.07291709 1.16763851]-[3 3]) = 2.0535411795004133
Norm_ij = norm([2.07291709 1.16763851]-[6 2]) = 4.0143250832738655
Norm_ij = norm([2.07291709 1.16763851]-[8 5]) = 7.058137608290476
distance: [2.0535411795004133, 4.0143250832738655, 7.058137608290476]
idx[197] = 0
Norm_ij = norm([0.94623208 0.24522253]-[3 3]) = 3.436096852065601
Norm_ij = norm([0.94623208 0.24522253]-[6 2]) = 5.349748978978818
Norm_ij = norm([0.94623208 0.24522253]-[8 5]) = 8.50667682691013
distance: [3.436096852065601, 5.349748978978818, 8.50667682691013]
idx[198] = 0
Norm_ij = norm([2.73911908 1.10072284]-[3 3]) = 1.9171104795567278
Norm_ij = norm([2.73911908 1.10072284]-[6 2]) = 3.382609024632832
Norm_ij = norm([2.73911908 1.10072284]-[8 5]) = 6.548376170923958
distance: [1.9171104795567278, 3.382609024632832, 6.548376170923958]
idx[199] = 0
Norm_ij = norm([6.00506534 2.72784171]-[3 3]) = 3.0173643811924418
Norm_ij = norm([6.00506534 2.72784171]-[6 2]) = 0.7278593392068636
Norm_ij = norm([6.00506534 2.72784171]-[8 5]) = 3.0236513672298666
distance: [3.0173643811924418, 0.7278593392068636, 3.0236513672298666]
idx[200] = 1
Norm_ij = norm([6.05696411 2.94970433]-[3 3]) = 3.05737783385421
Norm_ij = norm([6.05696411 2.94970433]-[6 2]) = 0.9514111776419146
Norm_ij = norm([6.05696411 2.94970433]-[8 5]) = 2.824730216697685
distance: [3.05737783385421, 0.9514111776419146, 2.824730216697685]
idx[201] = 1
Norm_ij = norm([6.77012767 3.21411422]-[3 3]) = 3.7762027912452467
Norm_ij = norm([6.77012767 3.21411422]-[6 2]) = 1.4377656106078078
Norm_ij = norm([6.77012767 3.21411422]-[8 5]) = 2.168403558348589
distance: [3.7762027912452467, 1.4377656106078078, 2.168403558348589]
idx[202] = 1
Norm_ij = norm([5.64034678 2.69385282]-[3 3]) = 2.6580363463490317
Norm_ij = norm([5.64034678 2.69385282]-[6 2]) = 0.7815255444327158
Norm_ij = norm([5.64034678 2.69385282]-[8 5]) = 3.2994360306426693
distance: [2.6580363463490317, 0.7815255444327158, 3.2994360306426693]
idx[203] = 1
Norm_ij = norm([5.63325403 2.99002339]-[3 3]) = 2.63327292654575
Norm_ij = norm([5.63325403 2.99002339]-[6 2]) = 1.0557693468714506
Norm_ij = norm([5.63325403 2.99002339]-[8 5]) = 3.1050752787148466
distance: [2.63327292654575, 1.0557693468714506, 3.1050752787148466]
idx[204] = 1
Norm_ij = norm([6.17443157 3.29026488]-[3 3]) = 3.187674617013247
Norm_ij = norm([6.17443157 3.29026488]-[6 2]) = 1.3020022398233462
Norm_ij = norm([6.17443157 3.29026488]-[8 5]) = 2.5011785799590553
distance: [3.187674617013247, 1.3020022398233462, 2.5011785799590553]
idx[205] = 1
Norm_ij = norm([7.24694794 2.96877424]-[3 3]) = 4.247062734154979
Norm_ij = norm([7.24694794 2.96877424]-[6 2]) = 1.579051202980019
Norm_ij = norm([7.24694794 2.96877424]-[8 5]) = 2.166325340729216
distance: [4.247062734154979, 1.579051202980019, 2.166325340729216]
idx[206] = 1
Norm_ij = norm([5.58162906 3.33510375]-[3 3]) = 2.6032869884922025
Norm_ij = norm([5.58162906 3.33510375]-[6 2]) = 1.3991198137689849
Norm_ij = norm([5.58162906 3.33510375]-[8 5]) = 2.9360513483796664
distance: [2.6032869884922025, 1.3991198137689849, 2.9360513483796664]
idx[207] = 1
Norm_ij = norm([5.3627205  3.14681192]-[3 3]) = 2.3672773225948203
Norm_ij = norm([5.3627205  3.14681192]-[6 2]) = 1.3119842758844187
Norm_ij = norm([5.3627205  3.14681192]-[8 5]) = 3.2232823628207004
distance: [2.3672773225948203, 1.3119842758844187, 3.2232823628207004]
idx[208] = 1
Norm_ij = norm([4.70775773 2.78710869]-[3 3]) = 1.7209762300985236
Norm_ij = norm([4.70775773 2.78710869]-[6 2]) = 1.5130863070239482
Norm_ij = norm([4.70775773 2.78710869]-[8 5]) = 3.96683085963628
distance: [1.7209762300985236, 1.5130863070239482, 3.96683085963628]
idx[209] = 1
Norm_ij = norm([7.42892098 3.4667949 ]-[3 3]) = 4.45345242832647
Norm_ij = norm([7.42892098 3.4667949 ]-[6 2]) = 2.0477554635062876
Norm_ij = norm([7.42892098 3.4667949 ]-[8 5]) = 1.636107922898012
distance: [4.45345242832647, 2.0477554635062876, 1.636107922898012]
idx[210] = 2
Norm_ij = norm([6.64107248 3.05998738]-[3 3]) = 3.6415666008922574
Norm_ij = norm([6.64107248 3.05998738]-[6 2]) = 1.2387684093740468
Norm_ij = norm([6.64107248 3.05998738]-[8 5]) = 2.368614144671538
distance: [3.6415666008922574, 1.2387684093740468, 2.368614144671538]
idx[211] = 1
Norm_ij = norm([6.37473652 2.56253059]-[3 3]) = 3.402973116726658
Norm_ij = norm([6.37473652 2.56253059]-[6 2]) = 0.6759202046341739
Norm_ij = norm([6.37473652 2.56253059]-[8 5]) = 2.929631121622233
distance: [3.402973116726658, 0.6759202046341739, 2.929631121622233]
idx[212] = 1
Norm_ij = norm([7.28780324 2.75179885]-[3 3]) = 4.294980846552002
Norm_ij = norm([7.28780324 2.75179885]-[6 2]) = 1.491187010756076
Norm_ij = norm([7.28780324 2.75179885]-[8 5]) = 2.3583113970774545
distance: [4.294980846552002, 1.491187010756076, 2.3583113970774545]
idx[213] = 1
Norm_ij = norm([6.20295231 2.67856179]-[3 3]) = 3.2190411646578014
Norm_ij = norm([6.20295231 2.67856179]-[6 2]) = 0.708262481433454
Norm_ij = norm([6.20295231 2.67856179]-[8 5]) = 2.935720655206322
distance: [3.2190411646578014, 0.708262481433454, 2.935720655206322]
```

```
idx[214] = 1
Norm_ij = norm([5.38736041 2.26737346]-[3 3]) = 2.4972447598795577
Norm_ij = norm([5.38736041 2.26737346]-[6 2]) = 0.6684428395536136
Norm_ij = norm([5.38736041 2.26737346]-[8 5]) = 3.7806260637271145
distance: [2.4972447598795577, 0.6684428395536136, 3.7806260637271145]
idx[215] = 1
Norm_ij = norm([5.6673103  2.96477867]-[3 3]) = 2.667542836014453
Norm_ij = norm([5.6673103  2.96477867]-[6 2]) = 1.0205294331922383
Norm_ij = norm([5.6673103  2.96477867]-[8 5]) = 3.095733690579456
distance: [2.667542836014453, 1.0205294331922383, 3.095733690579456]
idx[216] = 1
Norm_ij = norm([6.59702155 3.07082376]-[3 3]) = 3.5977187284481125
Norm_ij = norm([6.59702155 3.07082376]-[6 2]) = 1.2260090777483221
Norm_ij = norm([6.59702155 3.07082376]-[8 5]) = 2.3853866542625033
distance: [3.5977187284481125, 1.2260090777483221, 2.3853866542625033]
idx[217] = 1
Norm_ij = norm([7.75660559 3.15604465]-[3 3]) = 4.7591644966056075
Norm_ij = norm([7.75660559 3.15604465]-[6 2]) = 2.1028795580901716
Norm_ij = norm([7.75660559 3.15604465]-[8 5]) = 1.8599495111115192
distance: [4.7591644966056075, 2.1028795580901716, 1.8599495111115192]
idx[218] = 2
Norm_ij = norm([6.63262745 3.14799183]-[3 3]) = 3.635640767481576
Norm_ij = norm([6.63262745 3.14799183]-[6 2]) = 1.3107641767788352
Norm_ij = norm([6.63262745 3.14799183]-[8 5]) = 2.3020951237240475
distance: [3.635640767481576, 1.3107641767788352, 2.3020951237240475]
idx[219] = 1
Norm_ij = norm([5.76634959 3.14271707]-[3 3]) = 2.7700285567266754
Norm_ij = norm([5.76634959 3.14271707]-[6 2]) = 1.1663596475104023
Norm_ij = norm([5.76634959 3.14271707]-[8 5]) = 2.90494303380619
distance: [2.7700285567266754, 1.1663596475104023, 2.90494303380619]
idx[220] = 1
Norm_ij = norm([5.99423154 2.75707858]-[3 3]) = 3.0040694634382596
Norm_ij = norm([5.99423154 2.75707858]-[6 2]) = 0.757100560666345
Norm_ij = norm([5.99423154 2.75707858]-[8 5]) = 3.008953901422179
distance: [3.0040694634382596, 0.757100560666345, 3.008953901422179]
idx[221] = 1
Norm_ij = norm([6.37870407 2.65022321]-[3 3]) = 3.396760959445605
Norm_ij = norm([6.37870407 2.65022321]-[6 2]) = 0.7524672723798658
Norm_ij = norm([6.37870407 2.65022321]-[8 5]) = 2.8548294961421297
distance: [3.396760959445605, 0.7524672723798658, 2.8548294961421297]
idx[222] = 1
Norm_ij = norm([5.74036233 3.10391306]-[3 3]) = 2.7423317896525723
Norm_ij = norm([5.74036233 3.10391306]-[6 2]) = 1.134035168118332
Norm_ij = norm([5.74036233 3.10391306]-[8 5]) = 2.949764069427958
distance: [2.7423317896525723, 1.134035168118332, 2.949764069427958]
idx[223] = 1
Norm_ij = norm([4.61652442 2.79320715]-[3 3]) = 1.6296976632806277
Norm_ij = norm([4.61652442 2.79320715]-[6 2]) = 1.5947358002215146
Norm_ij = norm([4.61652442 2.79320715]-[8 5]) = 4.039534836929943
distance: [1.6296976632806277, 1.5947358002215146, 4.039534836929943]
idx[224] = 1
Norm_ij = norm([5.33533999 3.03928694]-[3 3]) = 2.3356704220412547
Norm_ij = norm([5.33533999 3.03928694]-[6 2]) = 1.233649174212422
Norm_ij = norm([5.33533999 3.03928694]-[8 5]) = 3.3082939213350193
distance: [2.3356704220412547, 1.233649174212422, 3.3082939213350193]
idx[225] = 1
Norm_ij = norm([5.37293912 2.81684776]-[3 3]) = 2.3799968139806515
Norm_ij = norm([5.37293912 2.81684776]-[6 2]) = 1.0297793925178633
Norm_ij = norm([5.37293912 2.81684776]-[8 5]) = 3.4157872538704788
distance: [2.3799968139806515, 1.0297793925178633, 3.4157872538704788]
idx[226] = 1
Norm_ij = norm([5.03611162 2.92486087]-[3 3]) = 2.037497582530783
Norm_ij = norm([5.03611162 2.92486087]-[6 2]) = 1.335832494419565
Norm_ij = norm([5.03611162 2.92486087]-[8 5]) = 3.618126138037762
distance: [2.037497582530783, 1.335832494419565, 3.618126138037762]
idx[227] = 1
Norm_ij = norm([5.52908677 3.33681576]-[3 3]) = 2.5514162220015217
Norm_ij = norm([5.52908677 3.33681576]-[6 2]) = 1.417333998000145
Norm_ij = norm([5.52908677 3.33681576]-[8 5]) = 2.9785221169862077
distance: [2.5514162220015217, 1.417333998000145, 2.9785221169862077]
idx[228] = 1
Norm_ij = norm([6.05086942 2.80702594]-[3 3]) = 3.0569663448725404
Norm_ij = norm([6.05086942 2.80702594]-[6 2]) = 0.8086275862484668
Norm_ij = norm([6.05086942 2.80702594]-[8 5]) = 2.9339811191795784
distance: [3.0569663448725404, 0.8086275862484668, 2.9339811191795784]
idx[229] = 1
Norm_ij = norm([5.132009   2.19812195]-[3 3]) = 2.2778214985868646
Norm_ij = norm([5.132009   2.19812195]-[6 2]) = 0.890314935933046
Norm_ij = norm([5.132009   2.19812195]-[8 5]) = 4.009475402409132
distance: [2.2778214985868646, 0.890314935933046, 4.009475402409132]
idx[230] = 1
Norm_ij = norm([5.73284945 2.87738132]-[3 3]) = 2.735598919878387
Norm_ij = norm([5.73284945 2.87738132]-[6 2]) = 0.9171517871099323
Norm_ij = norm([5.73284945 2.87738132]-[8 5]) = 3.1057175781248825
distance: [2.735598919878387, 0.9171517871099323, 3.1057175781248825]
idx[231] = 1
Norm_ij = norm([6.78110732 3.05676866]-[3 3]) = 3.7815334512226215
Norm_ij = norm([6.78110732 3.05676866]-[6 2]) = 1.314111352150276
Norm_ij = norm([6.78110732 3.05676866]-[8 5]) = 2.293871706614646
distance: [3.7815334512226215, 1.314111352150276, 2.293871706614646]
idx[232] = 1
Norm_ij = norm([6.44834449 3.35299225]-[3 3]) = 3.466364555159343
Norm_ij = norm([6.44834449 3.35299225]-[6 2]) = 1.4253423471412283
Norm_ij = norm([6.44834449 3.35299225]-[8 5]) = 2.262801219915561
distance: [3.466364555159343, 1.4253423471412283, 2.262801219915561]
idx[233] = 1
```

```
Norm_ij = norm([6.39941482 2.89756948]-[3 3]) = 3.400957677572694
Norm_ij = norm([6.39941482 2.89756948]-[6 2]) = 0.9824271868074147
Norm_ij = norm([6.39941482 2.89756948]-[8 5]) = 2.642363902474886
distance: [3.400957677572694, 0.9824271868074147, 2.642363902474886]
idx[234] = 1
Norm_ij = norm([5.86067925 2.99577129]-[3 3]) = 2.8606823726297694
Norm_ij = norm([5.86067925 2.99577129]-[6 2]) = 1.0054704015034115
Norm_ij = norm([5.86067925 2.99577129]-[8 5]) = 2.9314887032659436
distance: [2.8606823726297694, 1.0054704015034115, 2.9314887032659436]
idx[235] = 1
Norm_ij = norm([6.44765183 3.16560945]-[3 3]) = 3.4516270998399117
Norm_ij = norm([6.44765183 3.16560945]-[6 2]) = 1.2486142567472391
Norm_ij = norm([6.44765183 3.16560945]-[8 5]) = 2.403075844004094
distance: [3.4516270998399117, 1.2486142567472391, 2.403075844004094]
idx[236] = 1
Norm_ij = norm([5.36708111 3.19502552]-[3 3]) = 2.3751016676415566
Norm_ij = norm([5.36708111 3.19502552]-[6 2]) = 1.3522841070206435
Norm_ij = norm([5.36708111 3.19502552]-[8 5]) = 3.1922084471985186
distance: [2.3751016676415566, 1.3522841070206435, 3.1922084471985186]
idx[237] = 1
Norm_ij = norm([5.88735565 3.34615566]-[3 3]) = 2.9080313627972236
Norm_ij = norm([5.88735565 3.34615566]-[6 2]) = 1.3508603975637794
Norm_ij = norm([5.88735565 3.34615566]-[8 5]) = 2.682995943820175
distance: [2.9080313627972236, 1.3508603975637794, 2.682995943820175]
idx[238] = 1
Norm_ij = norm([3.96162465 2.72025046]-[3 3]) = 1.001489779690077
Norm_ij = norm([3.96162465 2.72025046]-[6 2]) = 2.161882230905104
Norm_ij = norm([3.96162465 2.72025046]-[8 5]) = 4.6374274532063895
distance: [1.001489779690077, 2.161882230905104, 4.6374274532063895]
idx[239] = 0
Norm_ij = norm([6.28438193 3.17360643]-[3 3]) = 3.2889669916500086
Norm_ij = norm([6.28438193 3.17360643]-[6 2]) = 1.2075699317109654
Norm_ij = norm([6.28438193 3.17360643]-[8 5]) = 2.5058050242973215
distance: [3.2889669916500086, 1.2075699317109654, 2.5058050242973215]
idx[240] = 1
Norm_ij = norm([4.20584789 2.81647368]-[3 3]) = 1.2197340031568897
Norm_ij = norm([4.20584789 2.81647368]-[6 2]) = 1.9711953418823727
Norm_ij = norm([4.20584789 2.81647368]-[8 5]) = 4.377599504552008
distance: [1.2197340031568897, 1.9711953418823727, 4.377599504552008]
idx[241] = 0
Norm_ij = norm([5.32615581 3.03314047]-[3 3]) = 2.32639187610215
Norm_ij = norm([5.32615581 3.03314047]-[6 2]) = 1.233468776524404
Norm_ij = norm([5.32615581 3.03314047]-[8 5]) = 3.3193341401008682
distance: [2.32639187610215, 1.233468776524404, 3.3193341401008682]
idx[242] = 1
Norm_ij = norm([7.17135204 3.4122727 ]-[3 3]) = 4.191675868958095
Norm_ij = norm([7.17135204 3.4122727 ]-[6 2]) = 1.8348241807351984
Norm_ij = norm([7.17135204 3.4122727 ]-[8 5]) = 1.790959358704803
```

```
distance: [4.191675868958095, 1.8348241807351984, 1.790959358704803]
idx[243] = 2
Norm_ij = norm([7.4949275  2.84018754]-[3 3]) = 4.497767588274626
Norm_ij = norm([7.4949275  2.84018754]-[6 2]) = 1.7148537375383899
Norm_ij = norm([7.4949275  2.84018754]-[8 5]) = 2.218082074601037
distance: [4.497767588274626, 1.7148537375383899, 2.218082074601037]
idx[244] = 1
Norm_ij = norm([7.39807241 3.48487031]-[3 3]) = 4.42471922023238
Norm_ij = norm([7.39807241 3.48487031]-[6 2]) = 2.039472064531361
Norm_ij = norm([7.39807241 3.48487031]-[8 5]) = 1.6303173948211942
distance: [4.42471922023238, 2.039472064531361, 1.6303173948211942]
idx[245] = 2
Norm_ij = norm([5.02432984 2.98683179]-[3 3]) = 2.0243726700143947
Norm_ij = norm([5.02432984 2.98683179]-[6 2]) = 1.3877208767187408
Norm_ij = norm([5.02432984 2.98683179]-[8 5]) = 3.5926952494766327
distance: [2.0243726700143947, 1.3877208767187408, 3.5926952494766327]
idx[246] = 1
Norm_ij = norm([5.31712478 2.81741356]-[3 3]) = 2.3243074384223914
Norm_ij = norm([5.31712478 2.81741356]-[6 2]) = 1.0651213500915735
Norm_ij = norm([5.31712478 2.81741356]-[8 5]) = 3.4585405876534354
distance: [2.3243074384223914, 1.0651213500915735, 3.4585405876534354]
idx[247] = 1
Norm_ij = norm([5.87655237 3.21661109]-[3 3]) = 2.8846964964270216
Norm_ij = norm([5.87655237 3.21661109]-[6 2]) = 1.2228580725392237
Norm_ij = norm([5.87655237 3.21661109]-[8 5]) = 2.7729958269877675
distance: [2.8846964964270216, 1.2228580725392237, 2.7729958269877675]
idx[248] = 1
Norm_ij = norm([6.03762833 2.68303512]-[3 3]) = 3.0541205967831893
Norm_ij = norm([6.03762833 2.68303512]-[6 2]) = 0.6840708002758209
Norm_ij = norm([6.03762833 2.68303512]-[8 5]) = 3.036318303889142
distance: [3.0541205967831893, 0.6840708002758209, 3.036318303889142]
idx[249] = 1
Norm_ij = norm([5.91280273 2.85631938]-[3 3]) = 2.9163442612367962
Norm_ij = norm([5.91280273 2.85631938]-[6 2]) = 0.8607474884973878
Norm_ij = norm([5.91280273 2.85631938]-[8 5]) = 2.991949042085569
distance: [2.9163442612367962, 0.8607474884973878, 2.991949042085569]
idx[250] = 1
Norm_ij = norm([6.69451358 2.89056083]-[3 3]) = 3.6961341349450114
Norm_ij = norm([6.69451358 2.89056083]-[6 2]) = 1.12935720706516
Norm_ij = norm([6.69451358 2.89056083]-[8 5]) = 2.480731427396451
distance: [3.6961341349450114, 1.12935720706516, 2.480731427396451]
idx[251] = 1
Norm_ij = norm([6.01017978 2.72401338]-[3 3]) = 3.0228051400006017
Norm_ij = norm([6.01017978 2.72401338]-[6 2]) = 0.7240849433041926
Norm_ij = norm([6.01017978 2.72401338]-[8 5]) = 3.023160531308392
distance: [3.0228051400006017, 0.7240849433041926, 3.023160531308392]
idx[252] = 1
Norm_ij = norm([6.92721968 3.19960026]-[3 3]) = 3.932288737039541
Norm_ij = norm([6.92721968 3.19960026]-[6 2]) = 1.516171862548883
Norm_ij = norm([6.92721968 3.19960026]-[8 5]) = 2.0957807243463793
distance: [3.932288737039541, 1.516171862548883, 2.0957807243463793]
idx[253] = 1
Norm_ij = norm([6.33559522 3.30864291]-[3 3]) = 3.349844162464706
Norm_ij = norm([6.33559522 3.30864291]-[6 2]) = 1.3509886831273994
Norm_ij = norm([6.33559522 3.30864291]-[8 5]) = 2.3729585066205297
distance: [3.349844162464706, 1.3509886831273994, 2.3729585066205297]
idx[254] = 1
Norm_ij = norm([6.24257071 2.79179269]-[3 3]) = 3.249248386691018
Norm_ij = norm([6.24257071 2.79179269]-[6 2]) = 0.8281160596748968
Norm_ij = norm([6.24257071 2.79179269]-[8 5]) = 2.8221866083786957
distance: [3.249248386691018, 0.8281160596748968, 2.8221866083786957]
idx[255] = 1
Norm_ij = norm([5.57812294 3.24766016]-[3 3]) = 2.5899910118426366
Norm_ij = norm([5.57812294 3.24766016]-[6 2]) = 1.3170558592264223
Norm_ij = norm([5.57812294 3.24766016]-[8 5]) = 2.9893449796957436
distance: [2.5899910118426366, 1.3170558592264223, 2.9893449796957436]
idx[256] = 1
Norm_ij = norm([6.40773863 2.67554951]-[3 3]) = 3.4231492316585457
Norm_ij = norm([6.40773863 2.67554951]-[6 2]) = 0.7890614213550763
Norm_ij = norm([6.40773863 2.67554951]-[8 5]) = 2.8175106684562463
distance: [3.4231492316585457, 0.7890614213550763, 2.8175106684562463]
idx[257] = 1
Norm_ij = norm([6.80029526 3.17579578]-[3 3]) = 3.8043591040256706
Norm_ij = norm([6.80029526 3.17579578]-[6 2]) = 1.4223108709620953
Norm_ij = norm([6.80029526 3.17579578]-[8 5]) = 2.1833489203800456
distance: [3.8043591040256706, 1.4223108709620953, 2.1833489203800456]
idx[258] = 1
Norm_ij = norm([7.21684033 2.72896575]-[3 3]) = 4.225541613297856
Norm_ij = norm([7.21684033 2.72896575]-[6 2]) = 1.4184820913242588
Norm_ij = norm([7.21684033 2.72896575]-[8 5]) = 2.4022771815809456
distance: [4.225541613297856, 1.4184820913242588, 2.4022771815809456]
idx[259] = 1
Norm_ij = norm([6.5110074  2.72731907]-[3 3]) = 3.521580306088191
Norm_ij = norm([6.5110074  2.72731907]-[6 2]) = 0.888887836659041
Norm_ij = norm([6.5110074  2.72731907]-[8 5]) = 2.7170163032247663
distance: [3.521580306088191, 0.888887836659041, 2.7170163032247663]
idx[260] = 1
Norm_ij = norm([4.60630534 3.329458  ]-[3 3]) = 1.6397437058834443
Norm_ij = norm([4.60630534 3.329458  ]-[6 2]) = 1.926095373528632
Norm_ij = norm([4.60630534 3.329458  ]-[8 5]) = 3.7825750499767907
distance: [1.6397437058834443, 1.926095373528632, 3.7825750499767907]
idx[261] = 0
Norm_ij = norm([7.65503226 2.87095628]-[3 3]) = 4.656820552108977
Norm_ij = norm([7.65503226 2.87095628]-[6 2]) = 1.8702129938028245
Norm_ij = norm([7.65503226 2.87095628]-[8 5]) = 2.156810117480784
distance: [4.656820552108977, 1.8702129938028245, 2.156810117480784]
```

```
idx[262] = 1
Norm_ij = norm([5.50295759 2.62924634]-[3 3]) = 2.5302677672249945
Norm_ij = norm([5.50295759 2.62924634]-[6 2]) = 0.8018741286316182
Norm_ij = norm([5.50295759 2.62924634]-[8 5]) = 3.4432097948975082
distance: [2.5302677672249945, 0.8018741286316182, 3.4432097948975082]
idx[263] = 1
Norm_ij = norm([6.63060699 3.01502301]-[3 3]) = 3.630638067351817
Norm_ij = norm([6.63060699 3.01502301]-[6 2]) = 1.1949631299610368
Norm_ij = norm([6.63060699 3.01502301]-[8 5]) = 2.4115080089032257
distance: [3.630638067351817, 1.1949631299610368, 2.4115080089032257]
idx[264] = 1
Norm_ij = norm([3.45928006 2.68478445]-[3 3]) = 0.5570449012675949
Norm_ij = norm([3.45928006 2.68478445]-[6 2]) = 2.6313850974350053
Norm_ij = norm([3.45928006 2.68478445]-[8 5]) = 5.096897153021857
distance: [0.5570449012675949, 2.6313850974350053, 5.096897153021857]
idx[265] = 0
Norm_ij = norm([8.20339815 2.41693495]-[3 3]) = 5.235963826511945
Norm_ij = norm([8.20339815 2.41693495]-[6 2]) = 2.242498199879229
Norm_ij = norm([8.20339815 2.41693495]-[8 5]) = 2.591060757833687
distance: [5.235963826511945, 2.242498199879229, 2.591060757833687]
idx[266] = 1
Norm_ij = norm([4.95679428 2.89776297]-[3 3]) = 1.959463255048515
Norm_ij = norm([4.95679428 2.89776297]-[6 2]) = 1.3763199234959735
Norm_ij = norm([4.95679428 2.89776297]-[8 5]) = 3.698716211711408
distance: [1.959463255048515, 1.3763199234959735, 3.698716211711408]
idx[267] = 1
Norm_ij = norm([5.37052667 2.44954813]-[3 3]) = 2.4335969167825
Norm_ij = norm([5.37052667 2.44954813]-[6 2]) = 0.7735180651931528
Norm_ij = norm([5.37052667 2.44954813]-[8 5]) = 3.6631864147104634
distance: [2.4335969167825, 0.7735180651931528, 3.6631864147104634]
idx[268] = 1
Norm_ij = norm([5.69797866 2.94977132]-[3 3]) = 2.6984461813018266
Norm_ij = norm([5.69797866 2.94977132]-[6 2]) = 0.9966355637855947
Norm_ij = norm([5.69797866 2.94977132]-[8 5]) = 3.0826514352691503
distance: [2.6984461813018266, 0.9966355637855947, 3.0826514352691503]
idx[269] = 1
Norm_ij = norm([6.27376271 2.24256036]-[3 3]) = 3.3602436054526024
Norm_ij = norm([6.27376271 2.24256036]-[6 2]) = 0.3657616027373908
Norm_ij = norm([6.27376271 2.24256036]-[8 5]) = 3.253208961145291
distance: [3.3602436054526024, 0.3657616027373908, 3.253208961145291]
idx[270] = 1
Norm_ij = norm([5.05274526 2.75692163]-[3 3]) = 2.0670873703446646
Norm_ij = norm([5.05274526 2.75692163]-[6 2]) = 1.2125270701808883
Norm_ij = norm([5.05274526 2.75692163]-[8 5]) = 3.7037428471288787
distance: [2.0670873703446646, 1.2125270701808883, 3.7037428471288787]
idx[271] = 1
Norm_ij = norm([6.88575584 2.88845269]-[3 3]) = 3.8873565863103674
Norm_ij = norm([6.88575584 2.88845269]-[6 2]) = 1.2545563320563633
Norm_ij = norm([6.88575584 2.88845269]-[8 5]) = 2.387503316342903
distance: [3.8873565863103674, 1.2545563320563633, 2.387503316342903]
idx[272] = 1
Norm_ij = norm([4.1877442  2.89283463]-[3 3]) = 1.1925689461930726
Norm_ij = norm([4.1877442  2.89283463]-[6 2]) = 2.0202536429415043
Norm_ij = norm([4.1877442  2.89283463]-[8 5]) = 4.355851259273681
distance: [1.1925689461930726, 2.0202536429415043, 4.355851259273681]
idx[273] = 0
Norm_ij = norm([5.97510328 3.0259191 ]-[3 3]) = 2.975216180750162
Norm_ij = norm([5.97510328 3.0259191 ]-[6 2]) = 1.0262211487452313
Norm_ij = norm([5.97510328 3.0259191 ]-[8 5]) = 2.8279324832605117
distance: [2.975216180750162, 1.0262211487452313, 2.8279324832605117]
idx[274] = 1
Norm_ij = norm([6.09457129 2.61867975]-[3 3]) = 3.1179763680538266
Norm_ij = norm([6.09457129 2.61867975]-[6 2]) = 0.6258660901141503
Norm_ij = norm([6.09457129 2.61867975]-[8 5]) = 3.0498105980630608
distance: [3.1179763680538266, 0.6258660901141503, 3.0498105980630608]
idx[275] = 1
Norm_ij = norm([5.72395697 3.04454219]-[3 3]) = 2.724321127393
Norm_ij = norm([5.72395697 3.04454219]-[6 2]) = 1.080401844970827
Norm_ij = norm([5.72395697 3.04454219]-[8 5]) = 3.0006977665965047
distance: [2.724321127393, 1.080401844970827, 3.0006977665965047]
idx[276] = 1
Norm_ij = norm([4.37249767 3.05488217]-[3 3]) = 1.3735945240435539
Norm_ij = norm([4.37249767 3.05488217]-[6 2]) = 1.9394690557794747
Norm_ij = norm([4.37249767 3.05488217]-[8 5]) = 4.116097240826729
distance: [1.3735945240435539, 1.9394690557794747, 4.116097240826729]
idx[277] = 0
Norm_ij = norm([6.29206262 2.77573856]-[3 3]) = 3.299692334195073
Norm_ij = norm([6.29206262 2.77573856]-[6 2]) = 0.8288973898260865
Norm_ij = norm([6.29206262 2.77573856]-[8 5]) = 2.804351807052696
distance: [3.299692334195073, 0.8288973898260865, 2.804351807052696]
idx[278] = 1
Norm_ij = norm([5.14533035 4.13225692]-[3 3]) = 2.425788126785487
Norm_ij = norm([5.14533035 4.13225692]-[6 2]) = 2.297167769440342
Norm_ij = norm([5.14533035 4.13225692]-[8 5]) = 2.9836415446794655
distance: [2.425788126785487, 2.297167769440342, 2.9836415446794655]
idx[279] = 1
Norm_ij = norm([6.5870565  3.37508345]-[3 3]) = 3.6066136424018578
Norm_ij = norm([6.5870565  3.37508345]-[6 2]) = 1.4951554574576393
Norm_ij = norm([6.5870565  3.37508345]-[8 5]) = 2.1533144466021983
distance: [3.6066136424018578, 1.4951554574576393, 2.1533144466021983]
idx[280] = 1
Norm_ij = norm([5.78769095 3.29255127]-[3 3]) = 2.802999661973006
Norm_ij = norm([5.78769095 3.29255127]-[6 2]) = 1.309871720719361
Norm_ij = norm([5.78769095 3.29255127]-[8 5]) = 2.7945827000746215
distance: [2.802999661973006, 1.309871720719361, 2.7945827000746215]
idx[281] = 1
```

```
Norm_ij = norm([6.72798098 3.0043983 ]-[3 3]) = 3.727983573285962
Norm_ij = norm([6.72798098 3.0043983 ]-[6 2]) = 1.2404725917421617
Norm_ij = norm([6.72798098 3.0043983 ]-[8 5]) = 2.366528794386275
distance: [3.727983573285962, 1.2404725917421617, 2.366528794386275]
idx[282] = 1
Norm_ij = norm([6.64078939 2.41068839]-[3 3]) = 3.6881750948447913
Norm_ij = norm([6.64078939 2.41068839]-[6 2]) = 0.7611018263210415
Norm_ij = norm([6.64078939 2.41068839]-[8 5]) = 2.9243782439285355
distance: [3.6881750948447913, 0.7611018263210415, 2.9243782439285355]
idx[283] = 1
Norm_ij = norm([6.23228878 2.72850902]-[3 3]) = 3.2436704659153803
Norm_ij = norm([6.23228878 2.72850902]-[6 2]) = 0.7646459780345058
Norm_ij = norm([6.23228878 2.72850902]-[8 5]) = 2.8782762598712712
distance: [3.2436704659153803, 0.7646459780345058, 2.8782762598712712]
idx[284] = 1
Norm_ij = norm([6.21772724 2.80994633]-[3 3]) = 3.2233350699383974
Norm_ij = norm([6.21772724 2.80994633]-[6 2]) = 0.838700310633923
Norm_ij = norm([6.21772724 2.80994633]-[8 5]) = 2.8236202412979416
distance: [3.2233350699383974, 0.838700310633923, 2.8236202412979416]
idx[285] = 1
Norm_ij = norm([5.78116301 3.07987787]-[3 3]) = 2.782309861214551
Norm_ij = norm([5.78116301 3.07987787]-[6 2]) = 1.1018284057450278
Norm_ij = norm([5.78116301 3.07987787]-[8 5]) = 2.934298313428387
distance: [2.782309861214551, 1.1018284057450278, 2.934298313428387]
idx[286] = 1
Norm_ij = norm([6.62447253 2.74453743]-[3 3]) = 3.6334642231631578
Norm_ij = norm([6.62447253 2.74453743]-[6 2]) = 0.9717519854954809
Norm_ij = norm([6.62447253 2.74453743]-[8 5]) = 2.641815141085758
distance: [3.6334642231631578, 0.9717519854954809, 2.641815141085758]
idx[287] = 1
Norm_ij = norm([5.19590823 3.06972937]-[3 3]) = 2.197015052419959
Norm_ij = norm([5.19590823 3.06972937]-[6 2]) = 1.3382393242977795
Norm_ij = norm([5.19590823 3.06972937]-[8 5]) = 3.4042437303374733
distance: [2.197015052419959, 1.3382393242977795, 3.4042437303374733]
idx[288] = 1
Norm_ij = norm([5.87177181 3.2551773 ]-[3 3]) = 2.883086672465908
Norm_ij = norm([5.87177181 3.2551773 ]-[6 2]) = 1.2617101631722663
Norm_ij = norm([5.87177181 3.2551773 ]-[8 5]) = 2.7520467810713307
distance: [2.883086672465908, 1.2617101631722663, 2.7520467810713307]
idx[289] = 1
Norm_ij = norm([5.89562099 2.89843977]-[3 3]) = 2.8974014946724838
Norm_ij = norm([5.89562099 2.89843977]-[6 2]) = 0.904482727784579
Norm_ij = norm([5.89562099 2.89843977]-[8 5]) = 2.974048819793822
distance: [2.8974014946724838, 0.904482727784579, 2.974048819793822]
idx[290] = 1
Norm_ij = norm([5.6175432 2.5975071]-[3 3]) = 2.648307558696496
Norm_ij = norm([5.6175432 2.5975071]-[6 2]) = 0.7094278989713028
Norm_ij = norm([5.6175432 2.5975071]-[8 5]) = 3.3835000416859815
```

```
distance: [2.648307558696496, 0.7094278989713028, 3.3835000416859815]
idx[291] = 1
Norm_ij = norm([5.63176103 3.04758747]-[3 3]) = 2.6321912339037548
Norm_ij = norm([5.63176103 3.04758747]-[6 2]) = 1.110423095300497
Norm_ij = norm([5.63176103 3.04758747]-[8 5]) = 3.069278497020756
distance: [2.6321912339037548, 1.110423095300497, 3.069278497020756]
idx[292] = 1
Norm_ij = norm([5.50258659 3.11869075]-[3 3]) = 2.50539959623488
Norm_ij = norm([5.50258659 3.11869075]-[6 2]) = 1.2242912659616507
Norm_ij = norm([5.50258659 3.11869075]-[8 5]) = 3.1267232396359828
distance: [2.50539959623488, 1.2242912659616507, 3.1267232396359828]
idx[293] = 1
Norm_ij = norm([6.48212628 2.5508514 ]-[3 3]) = 3.51097392723381
Norm_ij = norm([6.48212628 2.5508514 ]-[6 2]) = 0.7320403092838061
Norm_ij = norm([6.48212628 2.5508514 ]-[8 5]) = 2.8813659087124845
distance: [3.51097392723381, 0.7320403092838061, 2.8813659087124845]
idx[294] = 1
Norm_ij = norm([7.30278708 3.38015979]-[3 3]) = 4.3195483727752135
Norm_ij = norm([7.30278708 3.38015979]-[6 2]) = 1.8979186560463992
Norm_ij = norm([7.30278708 3.38015979]-[8 5]) = 1.7635158526734014
distance: [4.3195483727752135, 1.8979186560463992, 1.7635158526734014]
idx[295] = 2
Norm_ij = norm([6.99198434 2.98706729]-[3 3]) = 3.9920052894189837
Norm_ij = norm([6.99198434 2.98706729]-[6 2]) = 1.3994051455992216
Norm_ij = norm([6.99198434 2.98706729]-[8 5]) = 2.2512204858098093
distance: [3.9920052894189837, 1.3994051455992216, 2.2512204858098093]
idx[296] = 1
Norm_ij = norm([4.8255341  2.77961664]-[3 3]) = 1.8387886119944976
Norm_ij = norm([4.8255341  2.77961664]-[6 2]) = 1.4096709775310616
Norm_ij = norm([4.8255341  2.77961664]-[8 5]) = 3.8739303101326965
distance: [1.8387886119944976, 1.4096709775310616, 3.8739303101326965]
idx[297] = 1
Norm_ij = norm([6.11768055 2.85475655]-[3 3]) = 3.121061946188485
Norm_ij = norm([6.11768055 2.85475655]-[6 2]) = 0.8628194917554485
Norm_ij = norm([6.11768055 2.85475655]-[8 5]) = 2.85397897009587
distance: [3.121061946188485, 0.8628194917554485, 2.85397897009587]
idx[298] = 1
Norm_ij = norm([0.94048944 5.71556802]-[3 3]) = 3.408209711558119
Norm_ij = norm([0.94048944 5.71556802]-[6 2]) = 6.277267944235466
Norm_ij = norm([0.94048944 5.71556802]-[8 5]) = 7.0956836836667545
distance: [3.408209711558119, 6.277267944235466, 7.0956836836667545]
idx[299] = 0
First three elements in idx are: [0 2 1]
Norm_ij = norm([-1. -1.]-[-1 -1]) = 0.0
Norm_ij = norm([-1. -1.]-[2 2]) = 4.242640687119285
distance: [0.0, 4.242640687119285]
idx[0] = 0
Norm_ij = norm([-1.5 -1.5]-[-1 -1]) = 0.7071067811865476
Norm_ij = norm([-1.5 -1.5]-[2 2]) = 4.949747468305833
distance: [0.7071067811865476, 4.949747468305833]
idx[1] = 0
Norm_ij = norm([-1.5 -1. ]-[-1 -1]) = 0.5
Norm_ij = norm([-1.5 -1. ]-[2 2]) = 4.6097722286464435
distance: [0.5, 4.6097722286464435]
idx[2] = 0
Norm_ij = norm([2. 2.]-[-1 -1]) = 4.242640687119285
Norm_ij = norm([2. 2.]-[2 2]) = 0.0
distance: [4.242640687119285, 0.0]
idx[3] = 1
Norm_ij = norm([2.5 2.5]-[-1 -1]) = 4.949747468305833
Norm_ij = norm([2.5 2.5]-[2 2]) = 0.7071067811865476
distance: [4.949747468305833, 0.7071067811865476]
idx[4] = 1
Norm_ij = norm([2.  2.5]-[-1 -1]) = 4.6097722286464435
Norm_ij = norm([2.  2.5]-[2 2]) = 0.5
distance: [4.6097722286464435, 0.5]
idx[5] = 1
Norm_ij = norm([-1. -1.]-[2.5 2. ]) = 4.6097722286464435
Norm_ij = norm([-1. -1.]-[-1. -1.]) = 0.0
Norm_ij = norm([-1. -1.]-[-1.5  1. ]) = 2.0615528128088303
distance: [4.6097722286464435, 0.0, 2.0615528128088303]
idx[0] = 1
Norm_ij = norm([-1.5 -1.5]-[2.5 2. ]) = 5.315072906367325
Norm_ij = norm([-1.5 -1.5]-[-1. -1.]) = 0.7071067811865476
Norm_ij = norm([-1.5 -1.5]-[-1.5  1. ]) = 2.5
distance: [5.315072906367325, 0.7071067811865476, 2.5]
idx[1] = 1
Norm_ij = norm([-1.5  1. ]-[2.5 2. ]) = 4.123105625617661
Norm_ij = norm([-1.5  1. ]-[-1. -1.]) = 2.0615528128088303
Norm_ij = norm([-1.5  1. ]-[-1.5  1. ]) = 0.0
distance: [4.123105625617661, 2.0615528128088303, 0.0]
idx[2] = 2
Norm_ij = norm([-1.   1.5]-[2.5 2. ]) = 3.5355339059327378
Norm_ij = norm([-1.   1.5]-[-1. -1.]) = 2.5
Norm_ij = norm([-1.   1.5]-[-1.5  1. ]) = 0.7071067811865476
distance: [3.5355339059327378, 2.5, 0.7071067811865476]
idx[3] = 2
Norm_ij = norm([2.5 1.5]-[2.5 2. ]) = 0.5
Norm_ij = norm([2.5 1.5]-[-1. -1.]) = 4.301162633521313
Norm_ij = norm([2.5 1.5]-[-1.5  1. ]) = 4.031128874149275
distance: [0.5, 4.301162633521313, 4.031128874149275]
idx[4] = 0
Norm_ij = norm([2. 2.]-[2.5 2. ]) = 0.5
Norm_ij = norm([2. 2.]-[-1. -1.]) = 4.242640687119285
Norm_ij = norm([2. 2.]-[-1.5  1. ]) = 3.640054944640259
distance: [0.5, 4.242640687119285, 3.640054944640259]
```

```
idx[5] = 0
Norm_ij = norm([-1. -1.]-[2.5 2. ]) = 4.6097722286464435
Norm_ij = norm([-1. -1.]-[-1. -1.]) = 0.0
Norm_ij = norm([-1. -1.]-[-1.5  1. ]) = 2.0615528128088303
distance: [4.6097722286464435, 0.0, 2.0615528128088303]
idx[0] = 1
Norm_ij = norm([-1.5 -1.5]-[2.5 2. ]) = 5.315072906367325
Norm_ij = norm([-1.5 -1.5]-[-1. -1.]) = 0.7071067811865476
Norm_ij = norm([-1.5 -1.5]-[-1.5  1. ]) = 2.5
distance: [5.315072906367325, 0.7071067811865476, 2.5]
idx[1] = 1
Norm_ij = norm([-1.5  1. ]-[2.5 2. ]) = 4.123105625617661
Norm_ij = norm([-1.5  1. ]-[-1. -1.]) = 2.0615528128088303
Norm_ij = norm([-1.5  1. ]-[-1.5  1. ]) = 0.0
distance: [4.123105625617661, 2.0615528128088303, 0.0]
idx[2] = 2
Norm_ij = norm([-1.   1.5]-[2.5 2. ]) = 3.5355339059327378
Norm_ij = norm([-1.   1.5]-[-1. -1.]) = 2.5
Norm_ij = norm([-1.   1.5]-[-1.5  1. ]) = 0.7071067811865476
distance: [3.5355339059327378, 2.5, 0.7071067811865476]
idx[3] = 2
Norm_ij = norm([2.5 1.5]-[2.5 2. ]) = 0.5
Norm_ij = norm([2.5 1.5]-[-1. -1.]) = 4.301162633521313
Norm_ij = norm([2.5 1.5]-[-1.5  1. ]) = 4.031128874149275
distance: [0.5, 4.301162633521313, 4.031128874149275]
idx[4] = 0
Norm_ij = norm([-1.1 -1.7]-[2.5 2. ]) = 5.162363799656123
Norm_ij = norm([-1.1 -1.7]-[-1. -1.]) = 0.7071067811865475
Norm_ij = norm([-1.1 -1.7]-[-1.5  1. ]) = 2.7294688127912363
distance: [5.162363799656123, 0.7071067811865475, 2.7294688127912363]
idx[5] = 1
Norm_ij = norm([-1.6  1.2]-[2.5 2. ]) = 4.177319714841085
Norm_ij = norm([-1.6  1.2]-[-1. -1.]) = 2.280350850198276
Norm_ij = norm([-1.6  1.2]-[-1.5  1. ]) = 0.22360679774997896
distance: [4.177319714841085, 2.280350850198276, 0.22360679774997896]
idx[6] = 2
All tests passed!
```

**Expected Output**:

First three elements in idx are

[0 2 1]

### 1.2 Computing centroid means

Given assignments of every point to a centroid, the second phase of the algorithm recomputes, for each centroid, the mean of the points that were assigned to it.

### Exercise 2

Please complete the `compute_centroids` below to recompute the value for each centroid

- Specifically, for every centroid $\mu_k$ we set

  $$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x^{(i)}$$

  where

  - $C_k$ is the set of examples that are assigned to centroid $k$
  - $|C_k|$ is the number of examples in the set $C_k$

- Concretely, if two examples say $x^{(3)}$ and $x^{(5)}$ are assigned to centroid $k = 2$, then you should update $\mu_2 = \frac{1}{2}(x^{(3)} + x^{(5)})$.

If you get stuck, you can check out the hints presented after the cell below to help you with the implementation.

[8]:

```python
# UNQ_C2
# GRADED FUNCTION: compute_centroids

def compute_centroids(X, idx, K):
    """
    Returns the new centroids by computing the means of the
    data points assigned to each centroid.

    Args:
        X (ndarray):   (m, n) Data points
        idx (ndarray): (m,) Array containing index of closest centroid for each
                       example in X. Concretely, idx[i] contains the index of
                       the centroid closest to example i
        K (int):       number of centroids

    Returns:
        centroids (ndarray): (K, n) New centroids computed
    """

    # Useful variables
    m, n = X.shape

    # You need to return the following variables correctly
    centroids = np.zeros((K, n))

    ### START CODE HERE ###
    for k in range(K):
        points = X[idx==k]
        centroids[k] = np.mean(points, axis=0)

    ### END CODE HERE ##
    return centroids
```

Click for hints

- Here's how you can structure the overall implementation for this function "'python def compute_centroids(X, idx, K): # Useful variables m, n = X.shape

  ```
  # You need to return the following variables correctly
  centroids = np.zeros((K, n))

  ### START CODE HERE ###
  for k in range(K):
      points = # Your code here to get a list of all data points in X assigned to centroid
      centroids[k] = # Your code here to compute the mean of the points assigned
  ```

  ### END CODE HERE ##

  return centroids "'

  If you're still stuck, you can check the hints presented below to figure out how to calculate `points` and `centroids[k]`.

  Hint to calculate points Say we wanted to find all the values in X that were assigned to cluster k=0. That is, the corresponding value in idx for these examples is 0. In Python, we can do it as X[idx == 0]. Similarly, the points assigned to centroid k=1 are X[idx == 1]

  More hints to calculate points You can compute points as points = X[idx == k]

  Hint to calculate centroids[k] You can use np.mean to find the mean. Make sure to set the parameter axis=0

  More hints to calculate centroids[k] You can compute centroids[k] as centroids[k] = np.mean(points, axis = 0)

Now check your implementation by running the cell below

```
[9]: K = 3
centroids = compute_centroids(X, idx, K)

print("The centroids are:", centroids)

# UNIT TEST
compute_centroids_test(compute_centroids)
```

```
The centroids are: [[2.42830111 3.15792418]
 [5.81350331 2.63365645]
 [7.11938687 3.6166844 ]]
All tests passed!
```

**Expected Output**:

2.42830111 3.15792418

5.81350331 2.63365645

7.11938687 3.6166844

## 2 - K-means on a sample dataset

After you have completed the two functions (`find_closest_centroids` and `compute_centroids`) above, the next step is to run the K-means algorithm on a toy 2D dataset to help you understand how K-means works. * We encourage you to take a look at the function (`run_kMeans`) below to understand how it works. * Notice that the code calls the two functions you implemented in a loop.

When you run the code below, it will produce a visualization that steps through the progress of the algorithm at each iteration. * At the end, your figure should look like the one displayed in Figure 1. * The final centroids are the black X-marks in the middle of the colored clusters. * You can see how these centroids got to their final location by looking at the other X-marks connected to it.

**Note**: You do not need to implement anything for this part. Simply run the code provided below

[13]:

```
# You do not need to implement anything for this part

def run_kMeans(X, initial_centroids, max_iters=10, plot_progress=False):
    """
    Runs the K-Means algorithm on data matrix X, where each row of X
    is a single example
    """

    # Initialize values
    m, n = X.shape
    K = initial_centroids.shape[0]
    centroids = initial_centroids
    previous_centroids = centroids
    idx = np.zeros(m)
    plt.figure(figsize=(8, 6))

    # Run K-Means
    for i in range(max_iters):

        #Output progress
        print("K-Means iteration %d/%d" % (i, max_iters-1))

        # For each example in X, assign it to the closest centroid
        idx = find_closest_centroids(X, centroids)

        # Optionally plot progress
        if plot_progress:
            plot_progress_kMeans(X, centroids, previous_centroids, idx, K, i)
            previous_centroids = centroids

        # Given the memberships, compute new centroids
        centroids = compute_centroids(X, idx, K)
    plt.show()
    return centroids, idx
```

[14]:

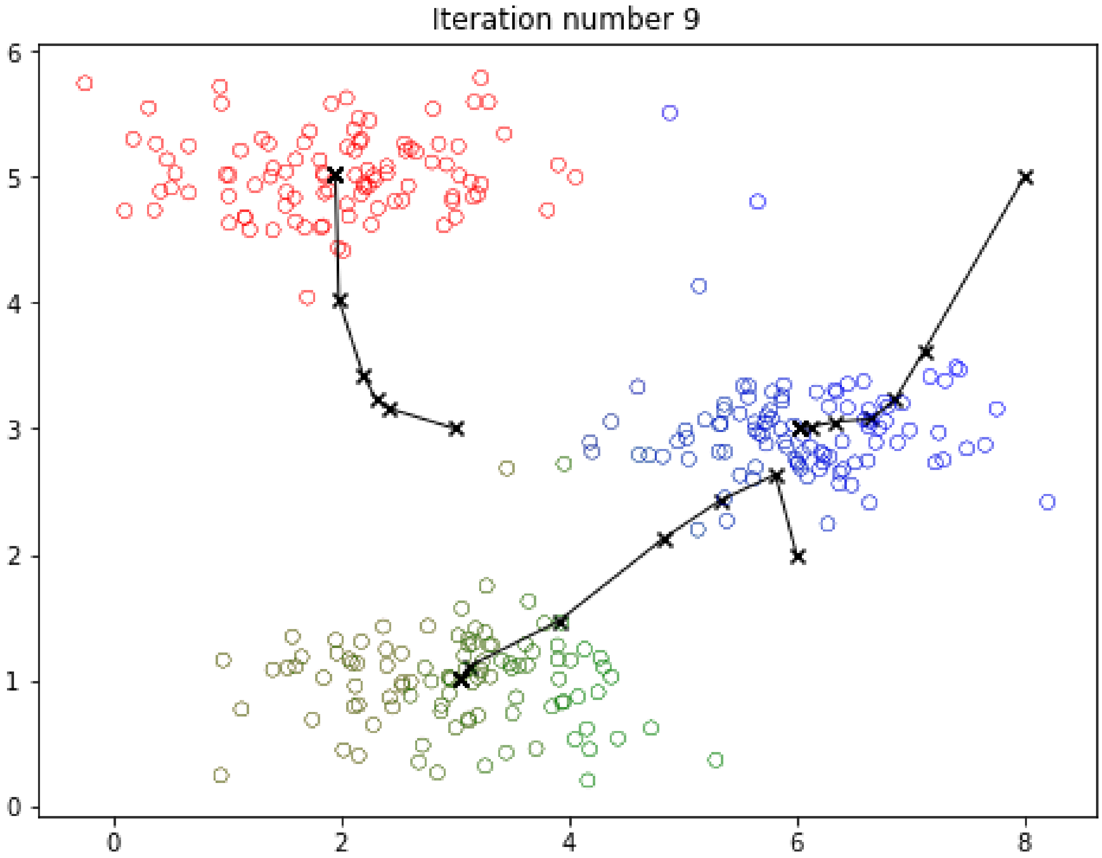
```
# Load an example dataset
X = load_data()

# Set initial centroids
initial_centroids = np.array([[3,3],[6,2],[8,5]])

# Number of iterations
max_iters = 10

# Run K-Means
centroids, idx = run_kMeans(X, initial_centroids, max_iters, plot_progress=True)
```

```
K-Means iteration 0/9
K-Means iteration 1/9
K-Means iteration 2/9
K-Means iteration 3/9
K-Means iteration 4/9
K-Means iteration 5/9
K-Means iteration 6/9
K-Means iteration 7/9
K-Means iteration 8/9
K-Means iteration 9/9
```



## 3 - Random initialization

The initial assignments of centroids for the example dataset was designed so that you will see the same figure as in Figure 1. In practice, a good strategy for initializing the centroids is to select random examples from the training set.

In this part of the exercise, you should understand how the function `kMeans_init_centroids` is implemented. * The code first randomly shuffles the indices of the examples (using `np.random.permutation()`). * Then, it selects the first $K$ examples based on the random permutation of the indices. * This allows the examples to be selected at random without the risk of selecting the same example twice.

**Note**: You do not need to implement anything for this part of the exercise.

```
[15]: # You do not need to modify this part

      def kMeans_init_centroids(X, K):
          """
          This function initializes K centroids that are to be
          used in K-Means on the dataset X
```

```
    Args:
        X (ndarray): Data points
        K (int):     number of centroids/clusters

    Returns:
        centroids (ndarray): Initialized centroids
    """

    # Randomly reorder the indices of examples
    randidx = np.random.permutation(X.shape[0])

    # Take the first K examples as centroids
    centroids = X[randidx[:K]]

    return centroids
```

You can run K-Means again but this time with random initial centroids. Run the cell below several times and observe how different clusters are created based on the initial points chosen.

[21]:

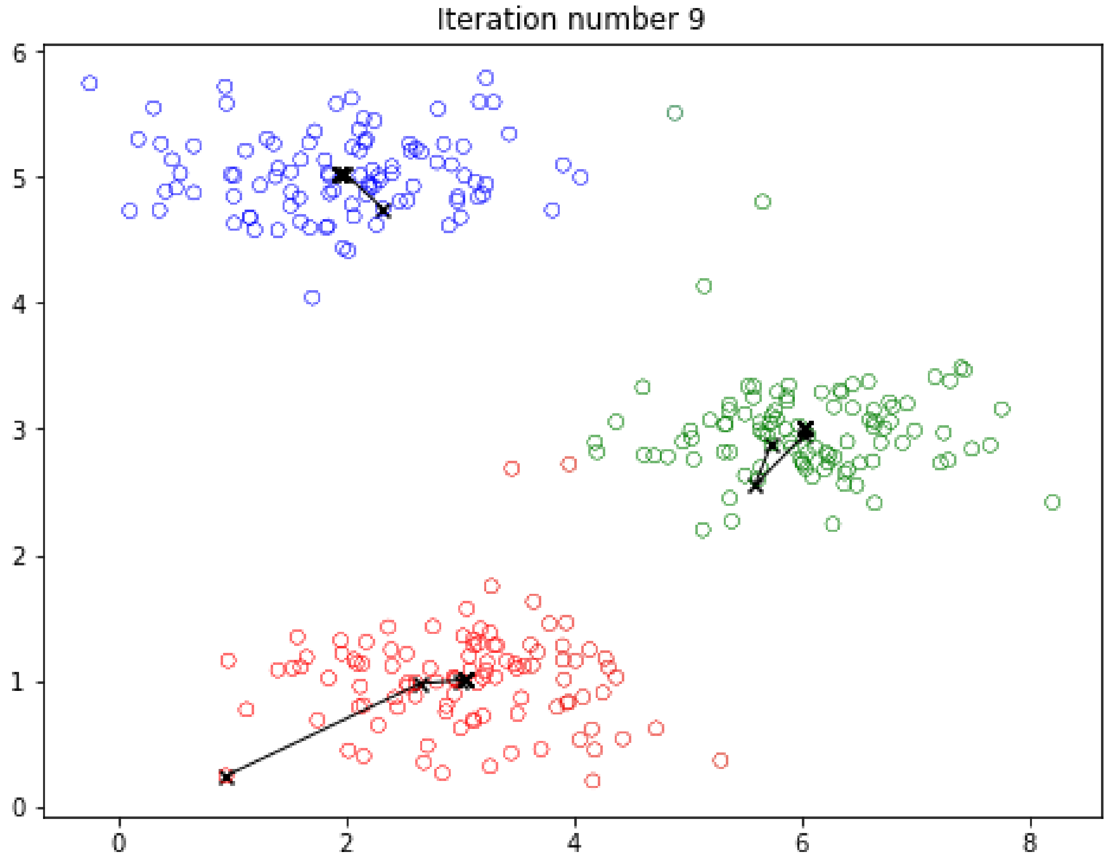
```
# Run this cell repeatedly to see different outcomes.

# Set number of centroids and max number of iterations
K = 3
max_iters = 10

# Set initial centroids by picking random examples from the dataset
initial_centroids = kMeans_init_centroids(X, K)

# Run K-Means
centroids, idx = run_kMeans(X, initial_centroids, max_iters, plot_progress=True)
```

```
K-Means iteration 0/9
K-Means iteration 1/9
K-Means iteration 2/9
K-Means iteration 3/9
K-Means iteration 4/9
K-Means iteration 5/9
K-Means iteration 6/9
K-Means iteration 7/9
K-Means iteration 8/9
K-Means iteration 9/9
```



## 4 - Image compression with K-means

In this exercise, you will apply K-means to image compression.

- In a straightforward 24-bit color representation of an image[2], each pixel is represented as three 8-bit unsigned integers (ranging from 0 to 255) that specify the red, green and blue intensity values. This encoding is often refered to as the RGB encoding.
- Our image contains thousands of colors, and in this part of the exercise, you will reduce the number of colors to 16 colors.
- By making this reduction, it is possible to represent (compress) the photo in an efficient way.
- Specifically, you only need to store the RGB values of the 16 selected colors, and for each pixel in the image you now need to only store the index of the color at that location (where only 4 bits are necessary to represent 16 possibilities).

In this part, you will use the K-means algorithm to select the 16 colors that will be used to represent the compressed image. * Concretely, you will treat every pixel in the original image as a data example and use the K-means algorithm to find the 16 colors that best group (cluster) the pixels in the 3- dimensional RGB space. * Once you have computed the cluster centroids on the image, you will then use the 16 colors to replace the pixels in the original image.

[2] The provided photo used in this exercise belongs to Frank Wouters and is used with his permission.

### 4.1 Dataset

**Load image**

First, you will use `matplotlib` to read in the original image, as shown below.

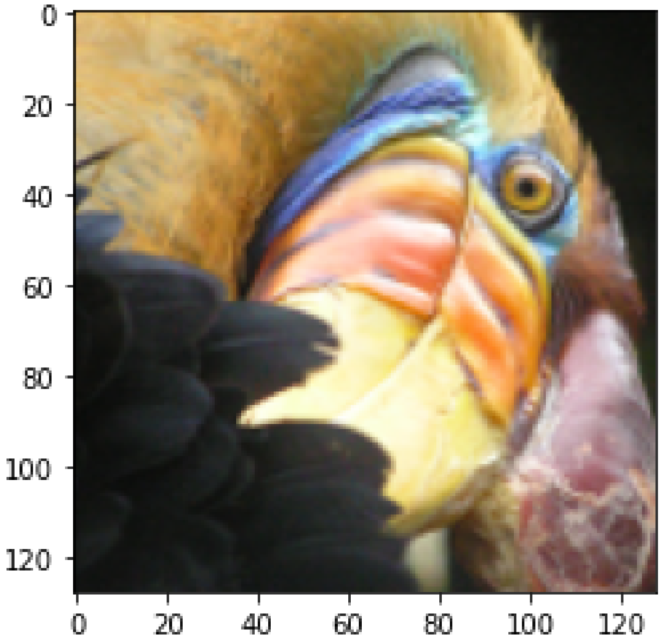
[22]:
```
# Load an image of a bird
original_img = plt.imread('bird_small.png')
```

**Visualize image**

You can visualize the image that was just loaded using the code below.

[23]:
```
# Visualizing the image
plt.imshow(original_img)
```

[23]:
```
<matplotlib.image.AxesImage at 0x7f925890f650>
```



**Check the dimension of the variable**

As always, you will print out the shape of your variable to get more familiar with the data.

[24]:
```
print("Shape of original_img is:", original_img.shape)
```

```
Shape of original_img is: (128, 128, 3)
```

As you can see, this creates a three-dimensional matrix `original_img` where * the first two indices identify a pixel position, and * the third index represents red, green, or blue.

For example, `original_img[50, 33, 2]` gives the blue intensity of the pixel at row 50 and column 33.

**Processing data** To call the `run_kMeans`, you need to first transform the matrix `original_img` into a two-dimensional matrix.

- The code below reshapes the matrix `original_img` to create an $m \times 3$ matrix of pixel colors (where $m = 16384 = 128 \times 128$)

*Note: If you'll try this exercise later on a JPG file, you first need to divide the pixel values by 255 so it will be in the range 0 to 1. This is not necessary for PNG files (e.g.* `bird_small.png`*) because it is already loaded in the required range (as mentioned in the plt.imread() documentation). We commented a line below for this so you can just uncomment it later in case you want to try a different file.*

[25]:
```
# Divide by 255 so that all values are in the range 0 - 1 (not needed for PNG files)
# original_img = original_img / 255

# Reshape the image into an m x 3 matrix where m = number of pixels
# (in this case m = 128 x 128 = 16384)
# Each row will contain the Red, Green and Blue pixel values
# This gives us our dataset matrix X_img that we will use K-Means on.

X_img = np.reshape(original_img, (original_img.shape[0] * original_img.shape[1], 3))
```

### 4.2 K-Means on image pixels

Now, run the cell below to run K-Means on the pre-processed image.

[26]:
```
# Run your K-Means algorithm on this data
# You should try different values of K and max_iters here
K = 16
max_iters = 10

# Using the function you have implemented above.
initial_centroids = kMeans_init_centroids(X_img, K)

# Run K-Means - this can take a couple of minutes depending on K and max_iters
centroids, idx = run_kMeans(X_img, initial_centroids, max_iters)
```

```
K-Means iteration 0/9
K-Means iteration 1/9
K-Means iteration 2/9
K-Means iteration 3/9
K-Means iteration 4/9
K-Means iteration 5/9
K-Means iteration 6/9
```

```
K-Means iteration 7/9
K-Means iteration 8/9
K-Means iteration 9/9
```

```
<Figure size 576x432 with 0 Axes>
```

[27]:

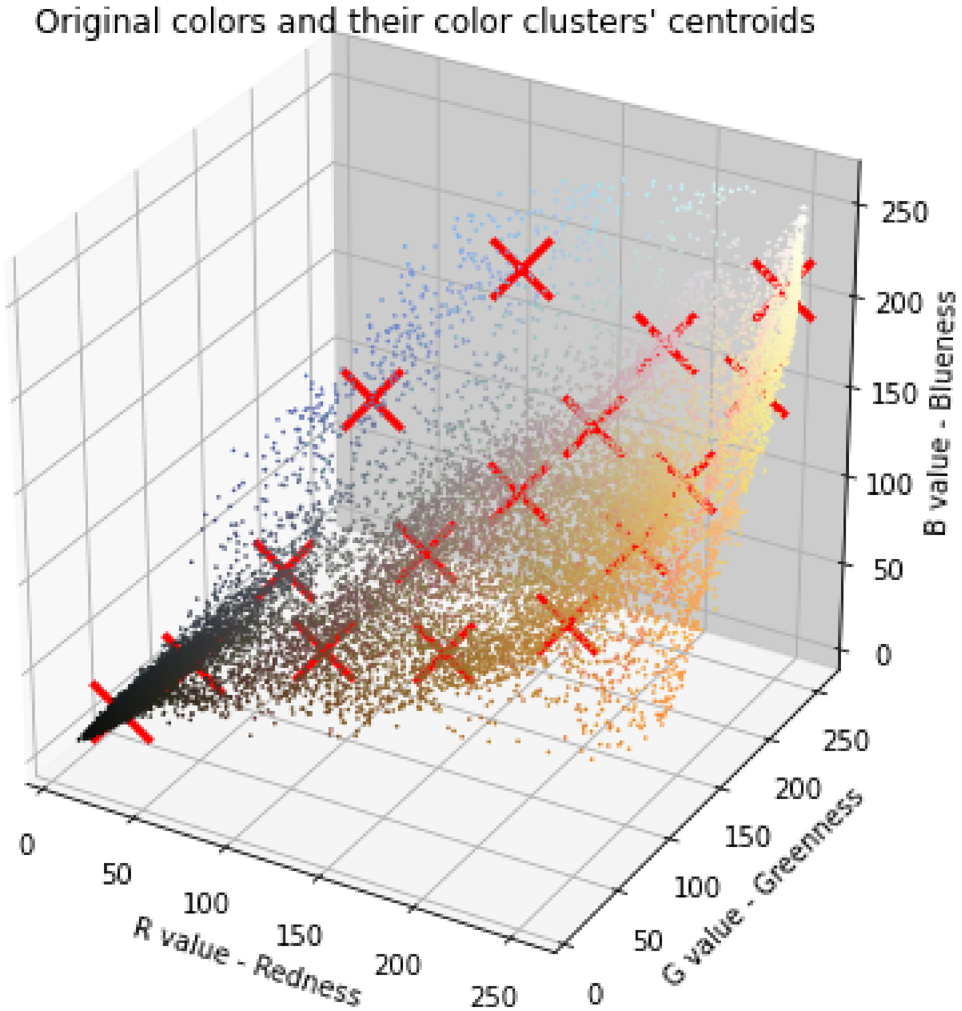
```
print("Shape of idx:", idx.shape)
print("Closest centroid for the first five elements:", idx[:5])
```

```
Shape of idx: (16384,)
Closest centroid for the first five elements: [ 2 11 11  2  2]
```

The code below will plot all the colors found in the original image. As mentioned earlier, the color of each pixel is represented by RGB values so the plot should have 3 axes – R, G, and B. You'll notice a lot of dots below representing thousands of colors in the original image. The red markers represent the centroids after running K-means. These will be the 16 colors that you will use to compress the image.

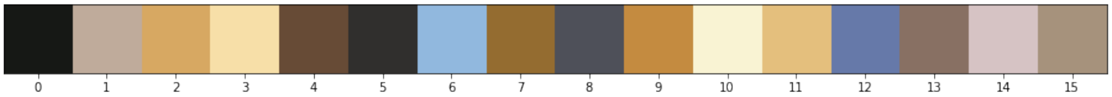
[28]:

```
# Plot the colors of the image and mark the centroids
plot_kMeans_RGB(X_img, centroids, idx, K)
```



You can visualize the colors at each of the red markers (i.e. the centroids) above with the function below. You will only see these colors when you generate the new image in the next section. The number below each color is its index and these are the numbers you see in the `idx` array.

[29]:
```
# Visualize the 16 colors selected
show_centroid_colors(centroids)
```



### 4.3 Compress the image

After finding the top $K = 16$ colors to represent the image, you can now assign each pixel position to its closest centroid using the `find_closest_centroids` function. * This allows you to represent the original image using the centroid assignments of each pixel. * Notice that you have significantly reduced the number of bits that are required to describe the image. * The original image required 24 bits (i.e. 8 bits x 3 channels in RGB encoding) for each one of the $128 \times 128$ pixel locations, resulting in total size of $128 \times 128 \times 24 = 393,216$ bits. * The new representation requires some overhead storage in form of a dictionary of 16 colors, each of which require 24 bits, but the image itself then only requires 4 bits per pixel location. * The final number of bits used is therefore $16 \times 24 + 128 \times 128 \times 4 = 65,920$ bits, which corresponds to compressing the original image by about a factor of 6.

[30]:

```
# Find the closest centroid of each pixel
idx = find_closest_centroids(X_img, centroids)

# Replace each pixel with the color of the closest centroid
X_recovered = centroids[idx, :]

# Reshape image into proper dimensions
X_recovered = np.reshape(X_recovered, original_img.shape)
```

Finally, you can view the effects of the compression by reconstructing the image based only on the centroid assignments. * Specifically, you replaced each pixel with the value of the centroid assigned to it. * Figure 3 shows a sample reconstruction. Even though the resulting image retains most of the characteristics of the original, you will also see some compression artifacts because of the fewer colors used.

- Run the code below to see how the image is reconstructed using the 16 colors selected earlier.

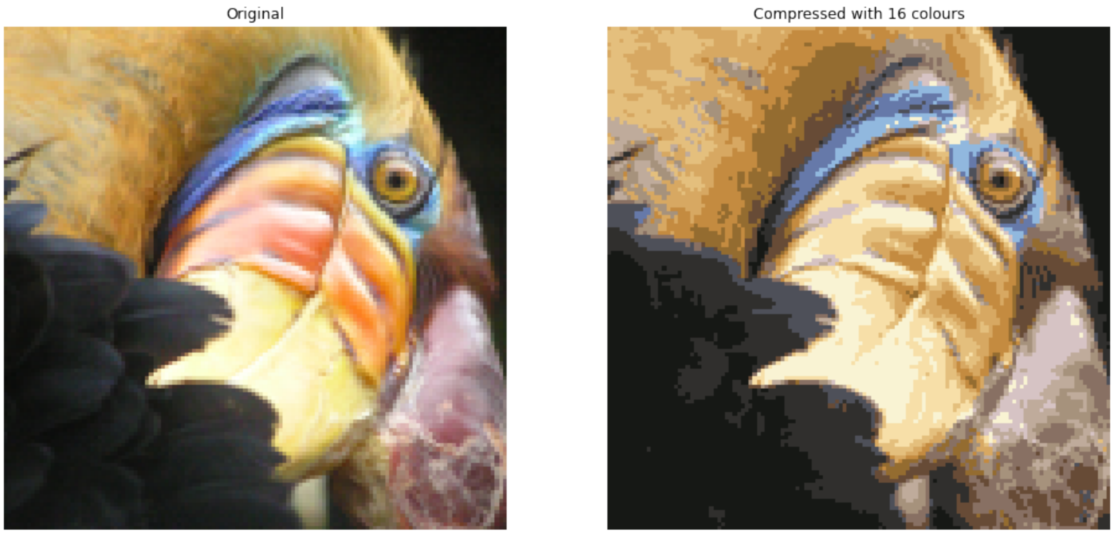
[31]:

```
# Display original image
fig, ax = plt.subplots(1,2, figsize=(16,16))
plt.axis('off')

ax[0].imshow(original_img)
ax[0].set_title('Original')
ax[0].set_axis_off()



# Display compressed image
ax[1].imshow(X_recovered)
ax[1].set_title('Compressed with %d colours'%K)
ax[1].set_axis_off()
```



**Congratulations on completing this lab on K-means clustering! In the next lesson, you will learn about another use case of unsupervised learning: anomaly detection. See you there!**

Please click here if you want to experiment with any of the non-graded code.

Important Note: Please only do this when you've already passed the assignment to avoid problems with the autograder.

On the notebook's menu, click "View" > "Cell Toolbar" > "Edit Metadata"

Hit the "Edit Metadata" button next to the code cell which you want to lock/unlock

Set the attribute value for "editable" to:

"true" if you want to unlock it

"false" if you want to lock it

```
    </li>
    <li> On the notebook's menu, click "View" > "Cell Toolbar" > "None" </li>
</ol>
<p> Here's a short demo of how to do the steps above:
    <br>
    <img src="https://drive.google.com/uc?export=view&id=14Xy_Mb17CZVgzVAgq7NCjMVBvSae3xO1" al
```