

Mathematics 3

Euclidean & Manhattan Distance



Name

Muhammad Baihaqi Aulia Asy'ari

NIM

2241720145

Class

2I

Department

Information Technology

Study Program

D4 Informatics Engineering

Task 1

Find the Euclidean distance from the center at (2,4)

1. (2,4) & (3,6)

$$\begin{aligned}d &= \sqrt{(3-2)^2 + (6-4)^2} \\&= \sqrt{(1)^2 + (2)^2} \\&= \sqrt{5} \\&= 2.23\end{aligned}$$

2. (2,4) & (5,3)

$$\begin{aligned}d &= \sqrt{(5-2)^2 + (3-4)^2} \\&= \sqrt{(3)^2 + (-1)^2} \\&= \sqrt{10} \\&= 3.16\end{aligned}$$

3. (2,4) & (7,1)

$$\begin{aligned}d &= \sqrt{(7-2)^2 + (1-4)^2} \\&= \sqrt{(5)^2 + (-3)^2} \\&= \sqrt{34} \\&= 5.83\end{aligned}$$

4. (2,4) & (6,8)

$$\begin{aligned}d &= \sqrt{(6-2)^2 + (8-4)^2} \\&= \sqrt{(4)^2 + (4)^2} \\&= 4\sqrt{2} \\&= 5.65\end{aligned}$$

Task 2

```
[ ] from scipy.spatial import distance
    p1 = (1, 2)
    p2 = (4, 5)
    d = distance.euclidean(p1,p2)
    print("Euclidean Distance: ", d)
```

```
Euclidean Distance:  4.242640687119285
```

Additional Task

1. A(2,3) & B(4,1)

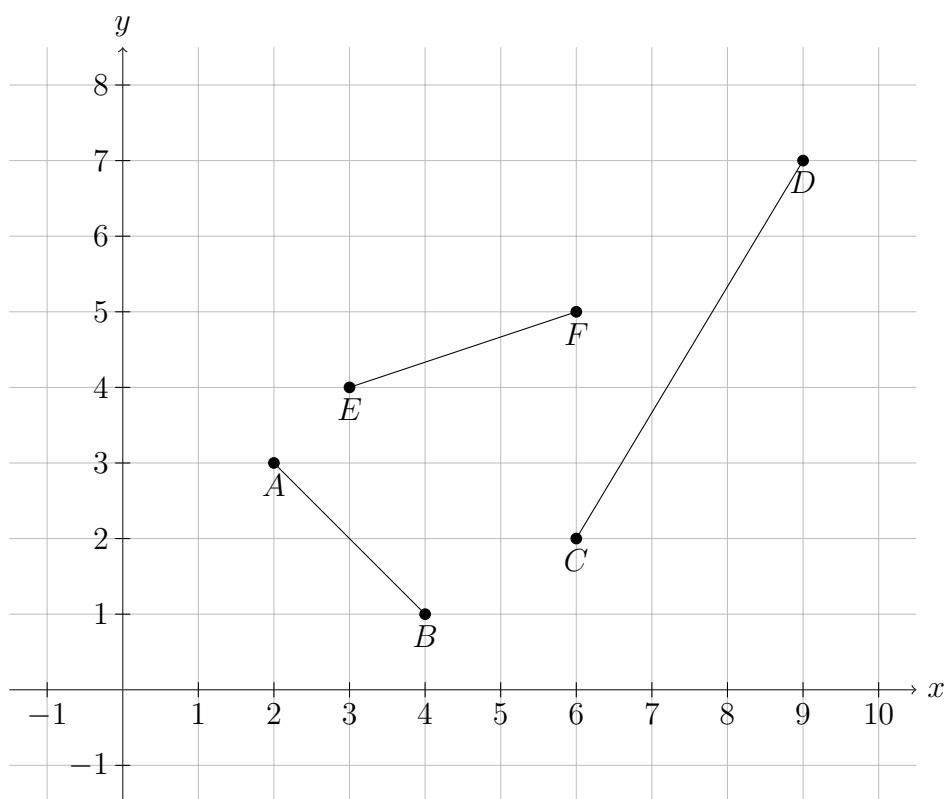
$$\begin{aligned}d &= \sqrt{(4-2)^2 + (1-3)^2} \\&= \sqrt{(2)^2 + (-2)^2} \\&= \sqrt{32} = 4\sqrt{2} \\&= 5.65685\end{aligned}$$

2. C(6,2) & D(9,7)

$$\begin{aligned}d &= \sqrt{(9-6)^2 + (7-2)^2} \\&= \sqrt{(3)^2 + (5)^2} \\&= \sqrt{34} \\&= 5.83095\end{aligned}$$

3. E(3,4) & F(6,5)

$$\begin{aligned}d &= \sqrt{(6-3)^2 + (5-4)^2} \\&= \sqrt{(3)^2 + (1)^2} \\&= \sqrt{10} \\&= 3.16228\end{aligned}$$



Task 3

1. (2,4) & (3,5)

$$\begin{aligned}d_{cd} &= |3 - 2| + |5 - 4| \\&= |1| + |1| \\&= 2\end{aligned}$$

2. (2,4) & (5,3)

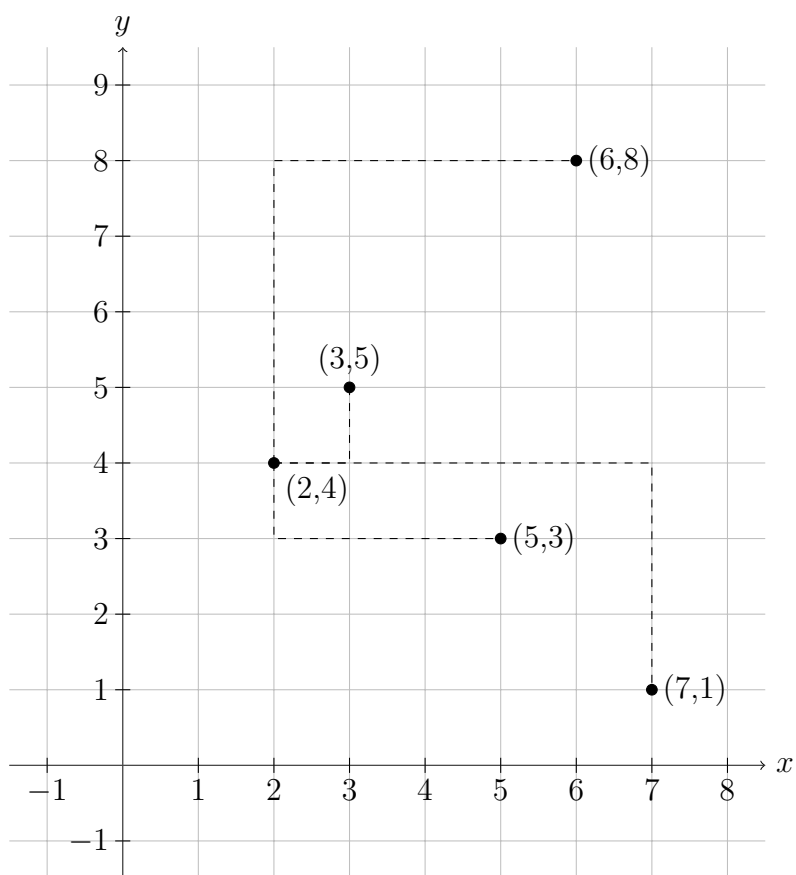
$$\begin{aligned}d_{cd} &= |5 - 2| + |3 - 4| \\&= |3| + |-1| \\&= 4\end{aligned}$$

3. (2,4) & (7,1)

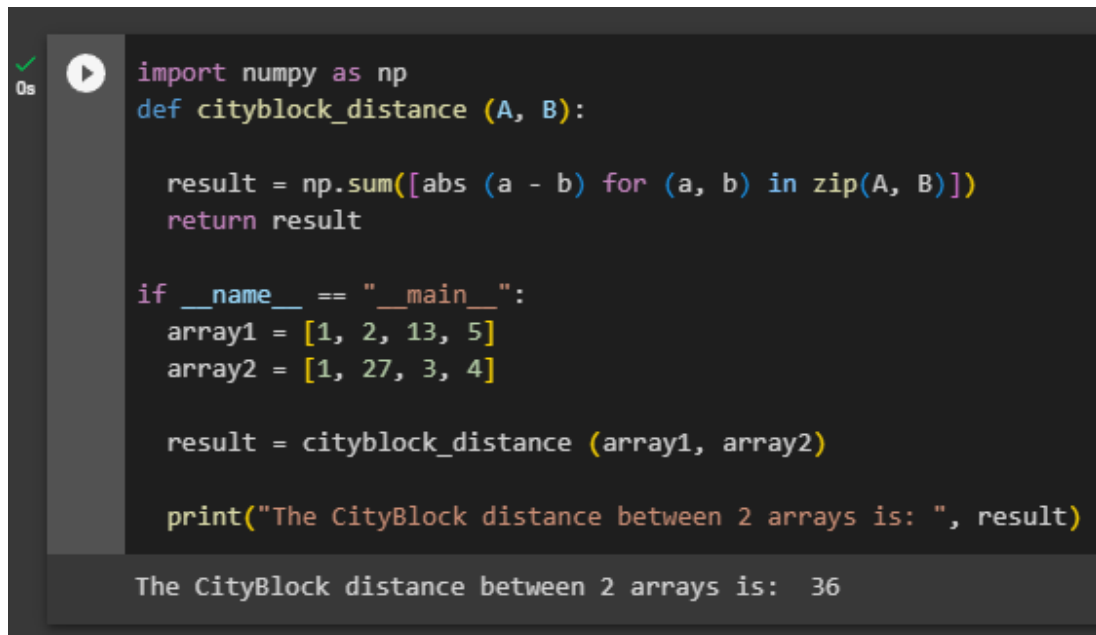
$$\begin{aligned}d_{cd} &= |7 - 2| + |1 - 4| \\&= |5| + |-3| \\&= 8\end{aligned}$$

4. (2,4) & (6,8)

$$\begin{aligned}d_{cd} &= |6 - 2| + |8 - 4| \\&= |4| + |4| \\&= 8\end{aligned}$$



Task 4



A screenshot of a Python code editor with a dark background. On the left, there is a vertical toolbar with a green checkmark and a play button icon. The code is written in a light-colored font. It defines a function `cityblock_distance` that takes two arrays `A` and `B` as input and returns the sum of absolute differences between corresponding elements. Below the function definition, there is a main block that creates two arrays, `array1` and `array2`, and prints the result of the `cityblock_distance` function applied to them. The output of the code is displayed at the bottom of the editor.

```
import numpy as np
def cityblock_distance (A, B):

    result = np.sum([abs (a - b) for (a, b) in zip(A, B)])
    return result

if __name__ == "__main__":
    array1 = [1, 2, 13, 5]
    array2 = [1, 27, 3, 4]

    result = cityblock_distance (array1, array2)

    print("The CityBlock distance between 2 arrays is: ", result)
```

The CityBlock distance between 2 arrays is: 36

Task 5

The Euclidean and City Block distance is method of measuring difference between two point to get the distance between the two point. The Euclidean distance measure the shortest distance in a straight line, while the City Block measure the absolute distance between two point from every axis or dimension it has.

The Euclidean formula goes as followed:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Adding more Variables as the dimension increase or for n-dimension it can be rewritten as

$$d = \sqrt{\sum_{i=1}^k (p_i - q_i)^2}$$

While the City Block formula goes as followed:

$$d_{cd} = |x_2 - x_1| + |y_2 - y_1| + |z_2 - z_1|$$

Adding more Variables as the dimension increase or for n-dimension it can be rewritten as

$$d_{cd} = \sum_{i=1}^k |p_i - q_i|$$

An applicative example of both method of measurement is for measuring shortest distance between two point. Other use of both method including but not limited to calculating the distance between two rows of data, clustering, and etc.