DBSCAN

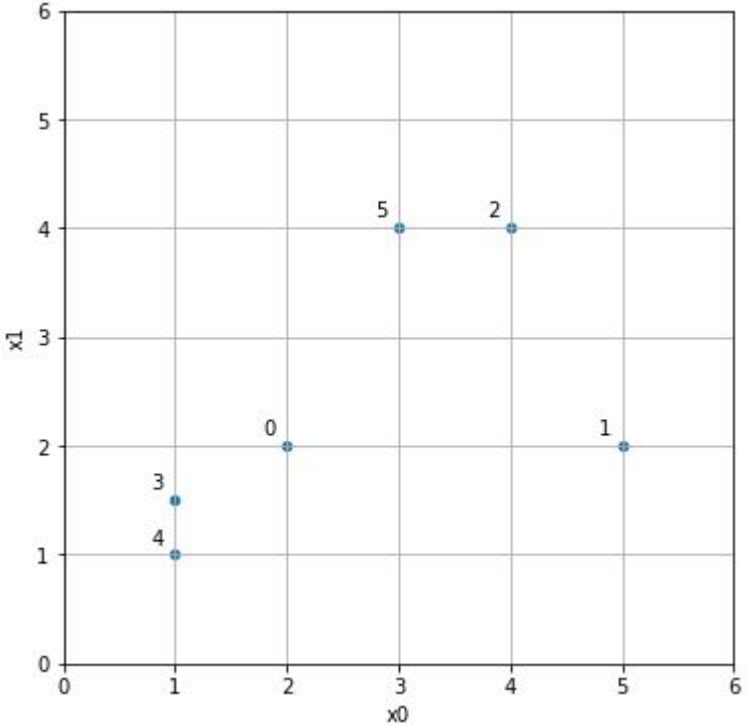
Data Mining Lab 6

Example:

The starting dataset:

- I did <u>not</u> standardize the data because I wanted to have nice numbers for the example, but **YOU HAVE TO**! (We are using distances ;))

	x0	x1
0	2.0	2.0
1	5.0	2.0
2	4.0	4.0
3	1.0	1.5
4	1.0	1.0
5	3.0	4.0

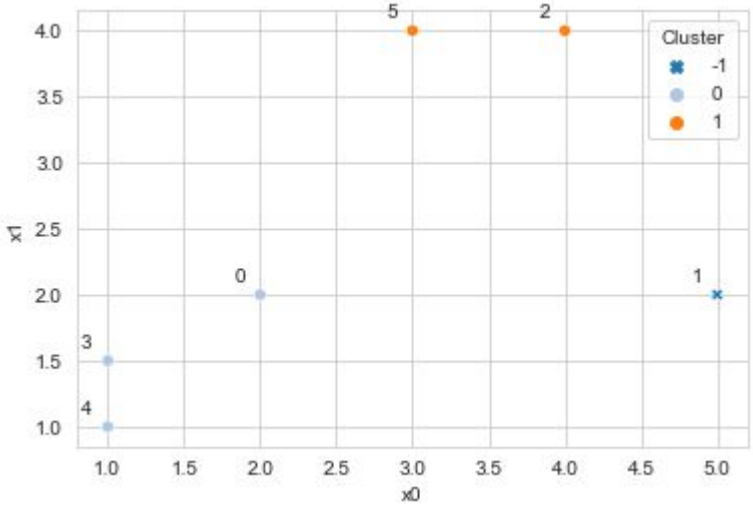


Final Result

Params:

- epsilon = 2.0
- min_samples = 2

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1



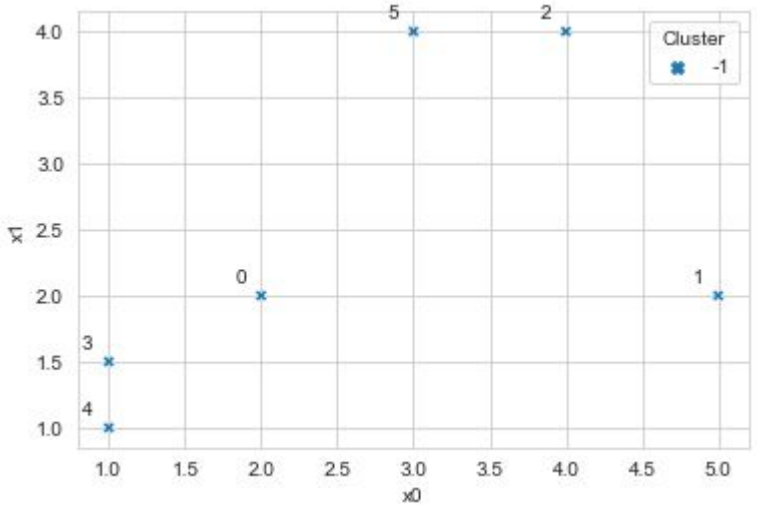
Initialization

Initialize clusters to -1 (= outlier)

visited = set()

queue = []

	x0	x1	Cluster
0	2.0	2.0	-1
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	-1
4	1.0	1.0	-1
5	3.0	4.0	-1

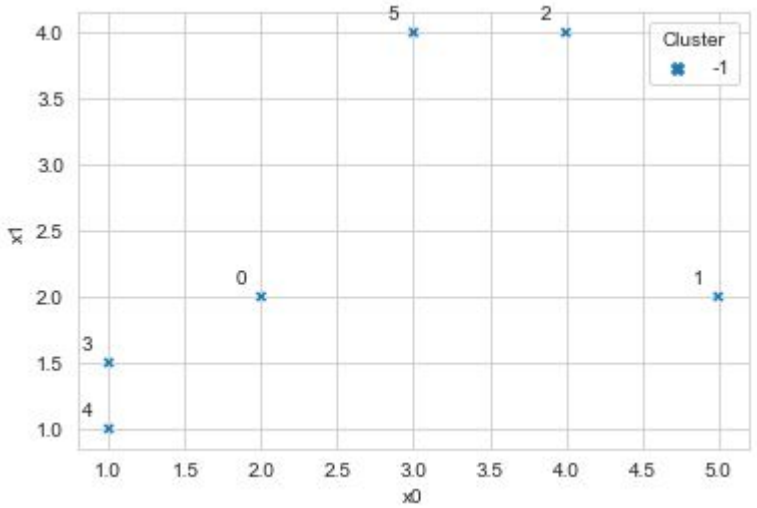


Step 0

Point 0 has not been visited yet. Start a cluster formation and add point 0 to the queue.

```
Step: 0
Current Cluster: 0 (from point 0)
Visited: {0}
Queue: [0]
```

	x0	x1	Cluster
0	2.0	2.0	-1
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	-1
4	1.0	1.0	-1
5	3.0	4.0	-1



Step 1

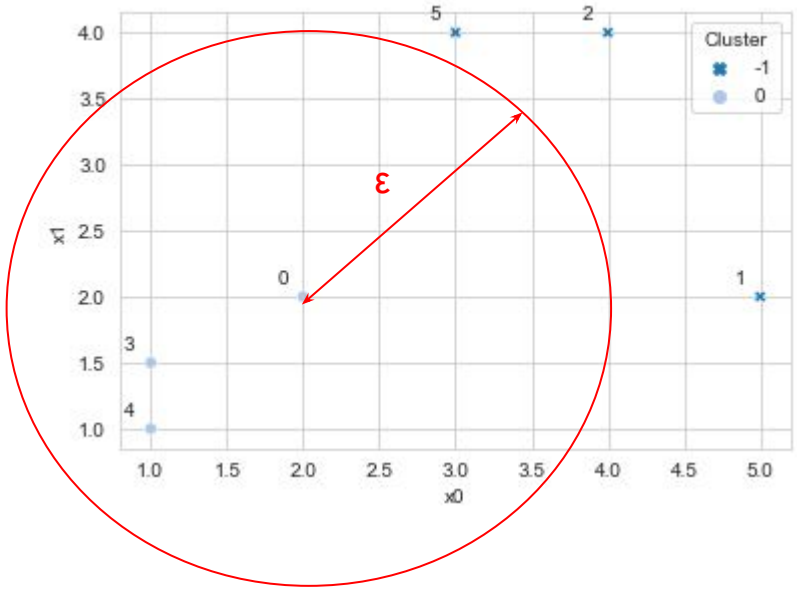
The queue is not empty -> pop point 0 from the queue.

Find its neighbourhood: [0, 3, 4]

Point 0 is a core point, so assign its neighbours to the current cluster (0), add the unvisited ones to the queue and set them as vi

```
Step: 1
Current Cluster: 0 (from point 0)
Visited: {0, 3, 4}
Queue: [3, 4]
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1



Step 2

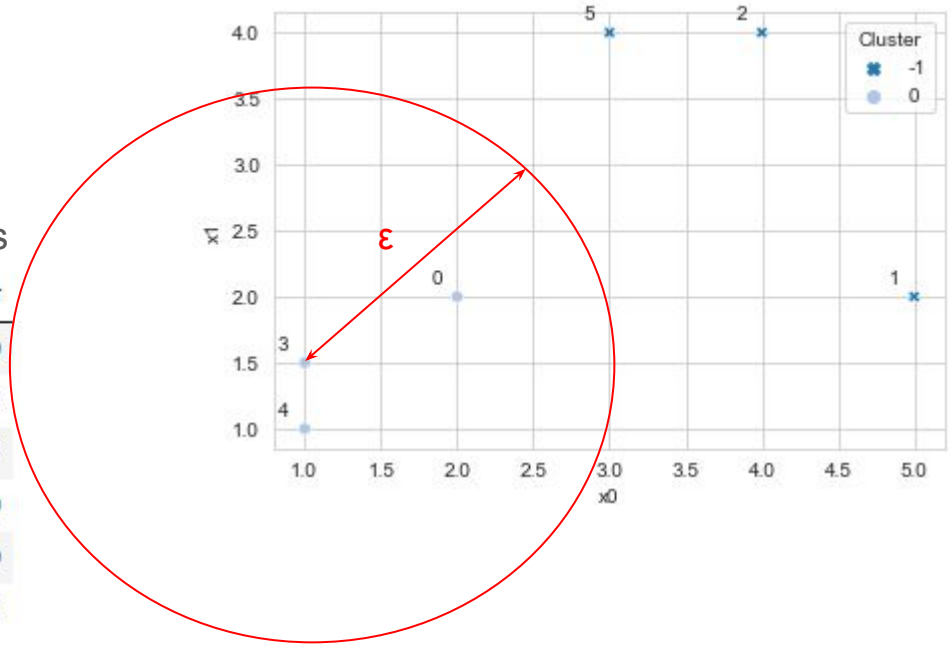
The queue is not empty -> pop point 3 from the queue.

Find its neighbourhood: [0, 3, 4]

Point 3 is a core point, so assign its neighbours to the current cluster (0), add the unvisited ones to the queue and set them as vi

```
Step: 2
Current Cluster: 0 (from point 0)
Visited: {0, 3, 4}
Queue: [4]
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1



Step 3

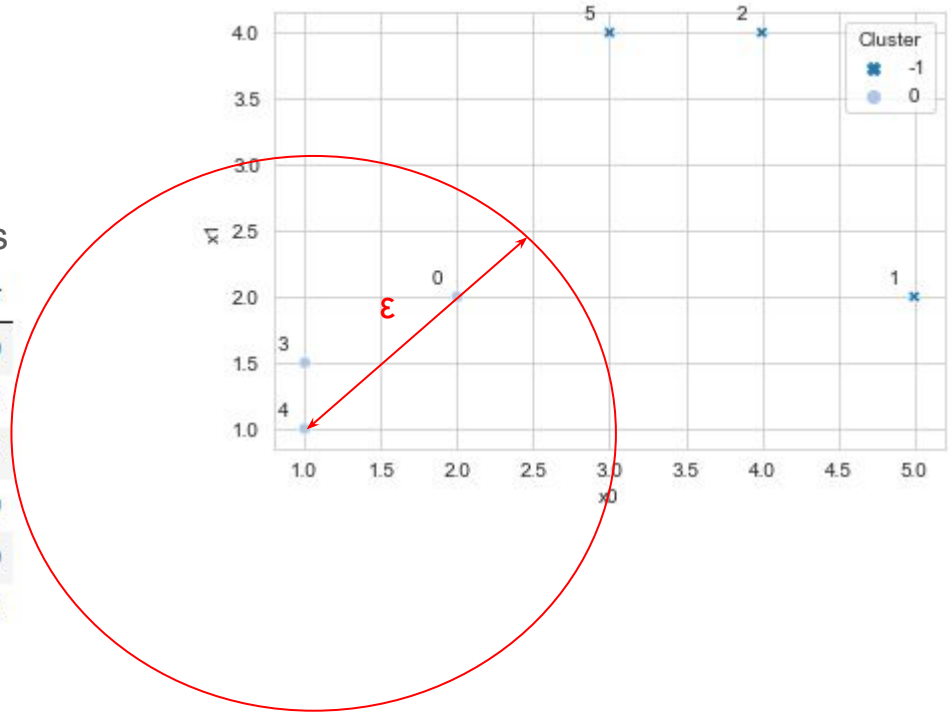
The queue is not empty -> pop point 4 from the queue.

Find its neighbourhood: [0, 3, 4]

Point 4 is a core point, so assign its neighbours to the current cluster (0), add the unvisited ones to the queue and set them as vi

```
Step: 3
Current Cluster: 0 (from point 0)
Visited: {0, 3, 4}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1

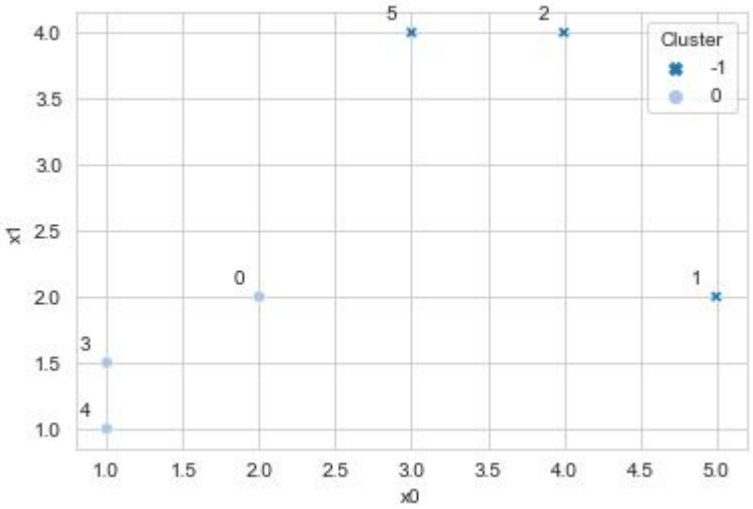


Step 4

The queue is empty and cluster 0 has been fully formed.

```
Step: 4
Current Cluster: 0 (from point 0)
Visited: {0, 3, 4}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1



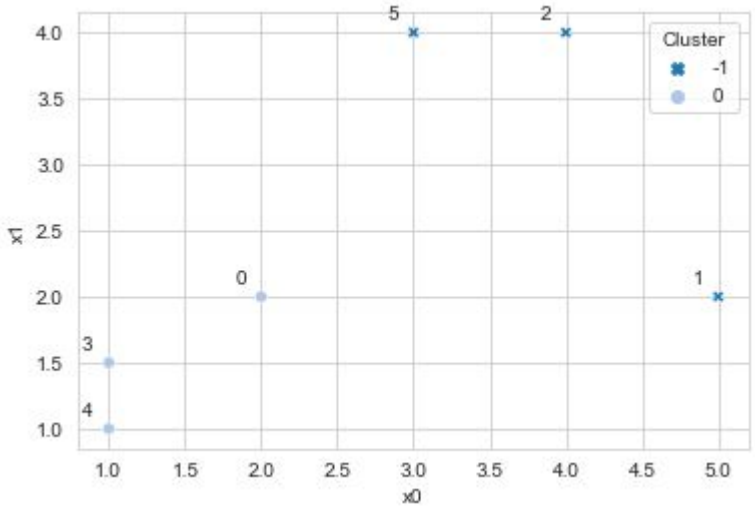
Step 5

Current Cluster++

Point 1 has not been visited yet. Start a cluster formation and add point 1 to the queue.

```
Step: 5
Current Cluster: 1 (from point 1)
Visited: {0, 1, 3, 4}
Queue: [1]
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1



Step 6

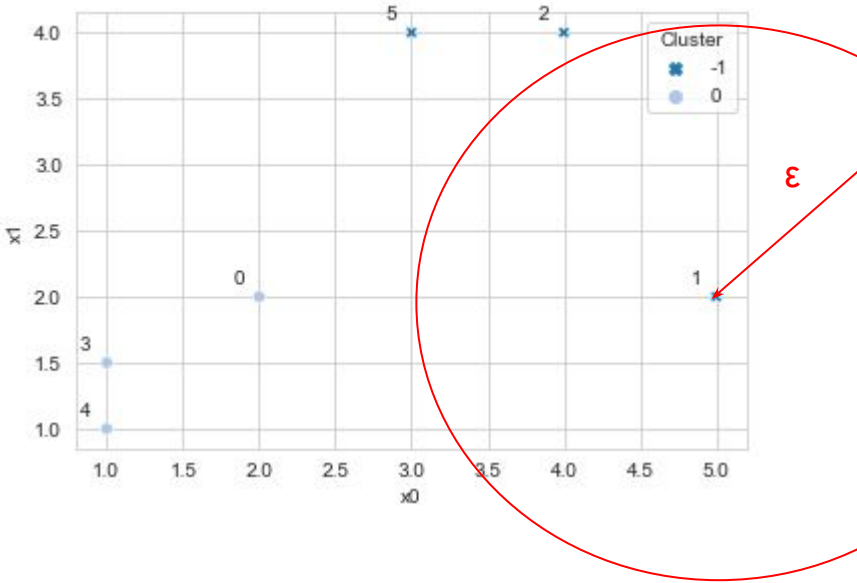
The queue is not empty -> pop point 1 from the queue.

Find its neighbourhood: [1]

Point 1 is not a core point, so just continue

```
Step: 6
Current Cluster: 1 (from point 1)
Visited: {0, 1, 3, 4}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1

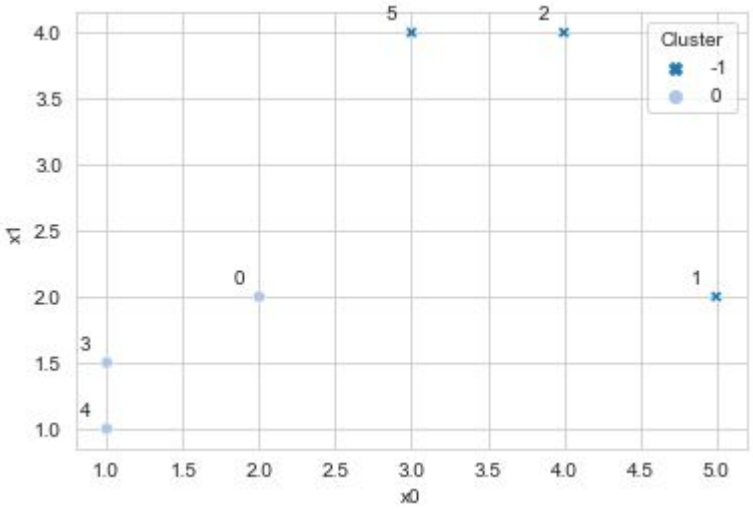


Step 7

The queue is empty and no cluster could be formed.

```
Step: 7
Current Cluster: 1 (from point 1)
Visited: {0, 1, 3, 4}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1



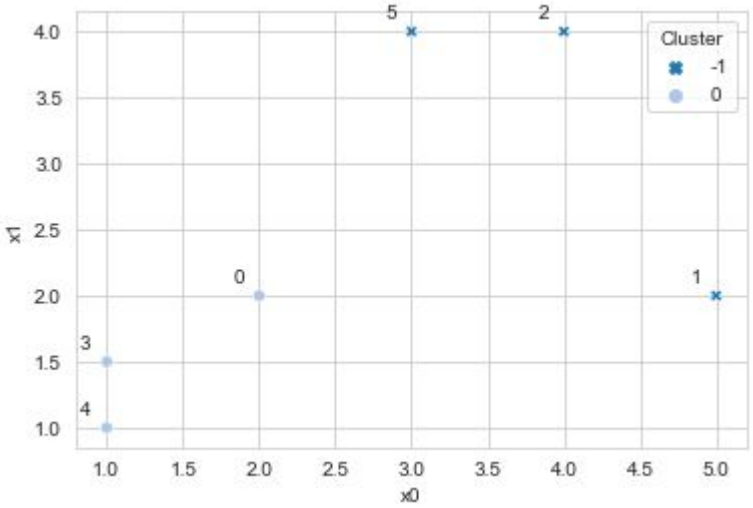
Step 8

Select the next point from the dataset.

Point 2 has not been visited yet. Start a cluster formation and add point 2 to the queue.

```
Step: 8
Current Cluster: 1 (from point 2)
Visited: {0, 1, 2, 3, 4}
Queue: [2]
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	-1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	-1



Step 9

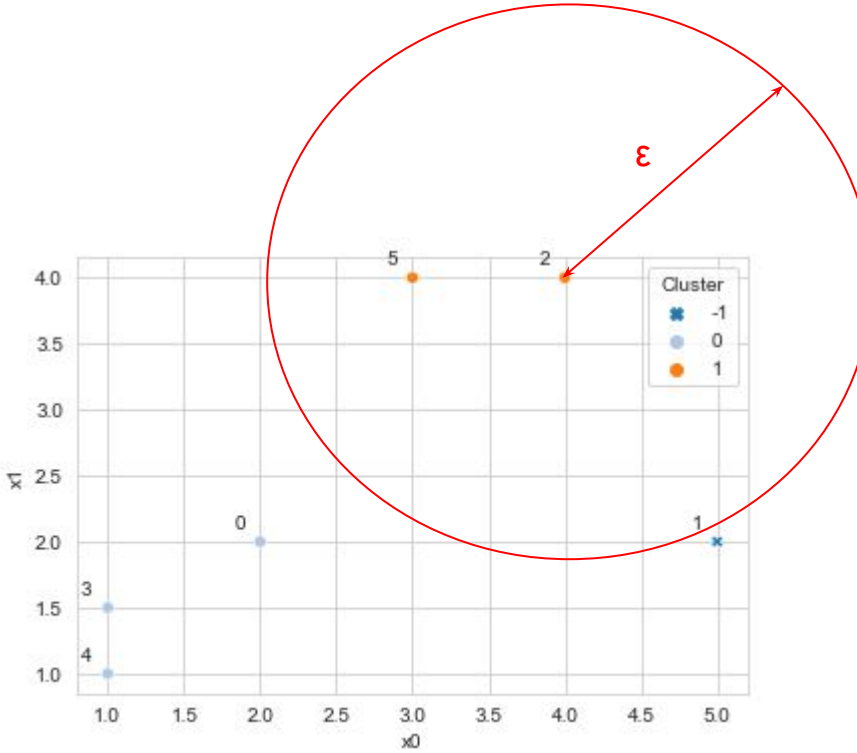
The queue is not empty -> pop point 2 from the queue.

Find its neighbourhood: [2, 5]

Point 2 is a core point, so assign its neighbours to the current cluster (1), add the unvisited ones to the queue and set them as visited

```
Step: 9
Current Cluster: 1 (from point 2)
Visited: {0, 1, 2, 3, 4, 5}
Queue: [5]
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1



Step 10

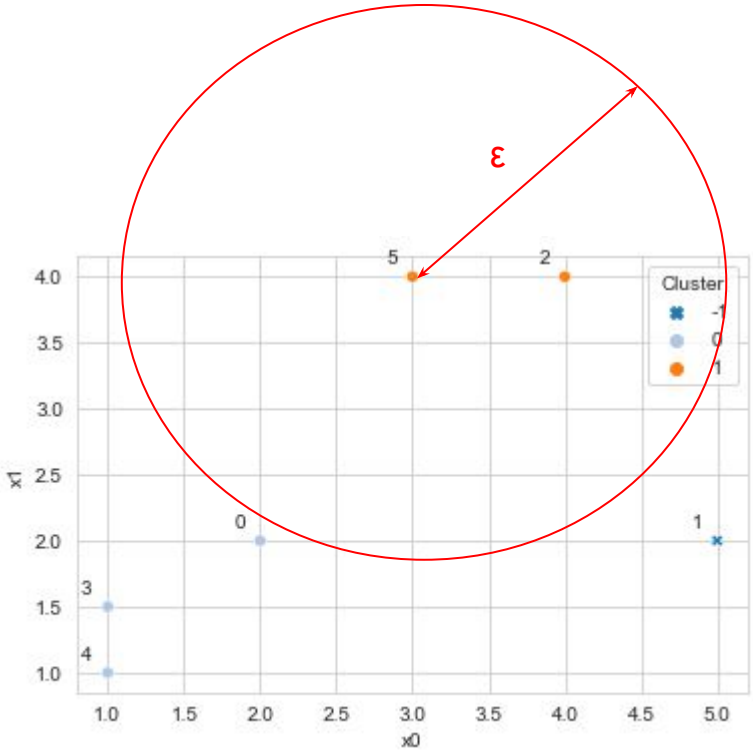
The queue is not empty -> pop point 5 from the queue.

Find its neighbourhood: [2, 5]

Point 5 is a core point, so assign its neighbours to the current cluster (1), add the unvisited ones to the queue and set them as visited

```
Step: 10
Current Cluster: 1 (from point 2)
Visited: {0, 1, 2, 3, 4, 5}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1

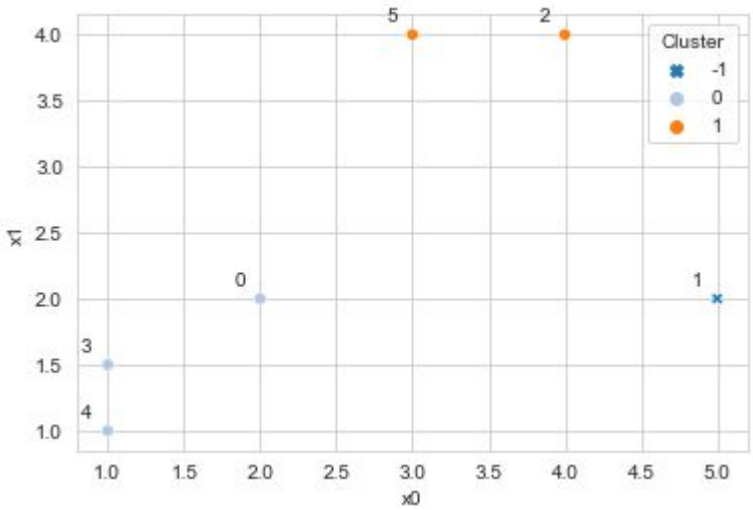


Step 11

The queue is empty and cluster 1 has been fully formed.

```
Step: 11
Current Cluster: 1 (from point 2)
Visited: {0, 1, 2, 3, 4, 5}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1



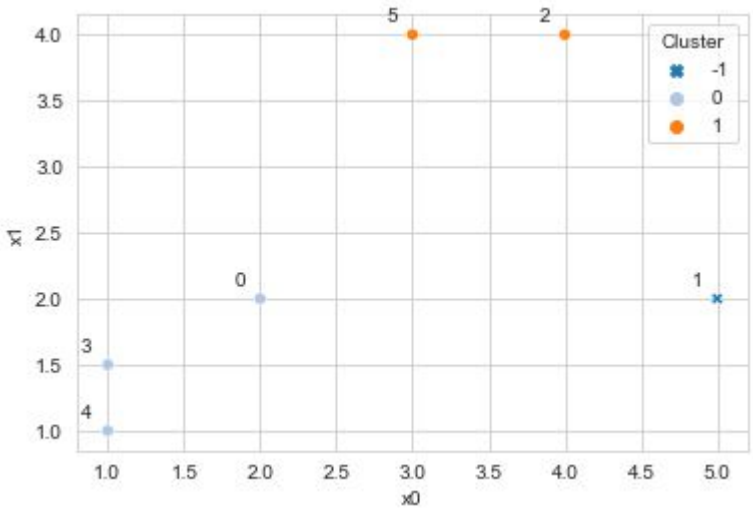
Step 12

Current Cluster++

Point 3 has already been visited -> continue.

```
Step: 12
Current Cluster: 2 (from point 3)
Visited: {0, 1, 2, 3, 4, 5}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1

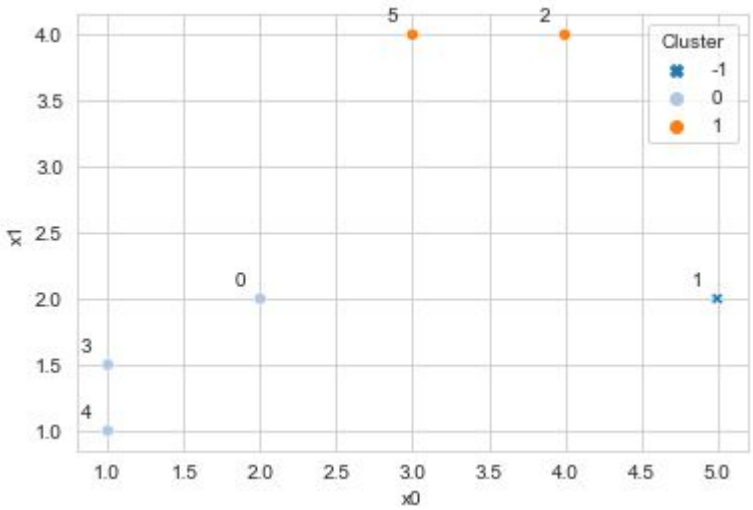


Step 13

Point 4 has already been visited -> continue.

```
Step: 13
Current Cluster: 2 (from point 4)
Visited: {0, 1, 2, 3, 4, 5}
Queue: []
```

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1



Step 14

Point 5 has already been visited -> All points have been visited -> **DONE**.

```
Step: 14
Current Cluster: 2 (from point
Visited: {0, 1, 2, 3, 4, 5}
Queue: []
```

5)

	x0	x1	Cluster
0	2.0	2.0	0
1	5.0	2.0	-1
2	4.0	4.0	1
3	1.0	1.5	0
4	1.0	1.0	0
5	3.0	4.0	1