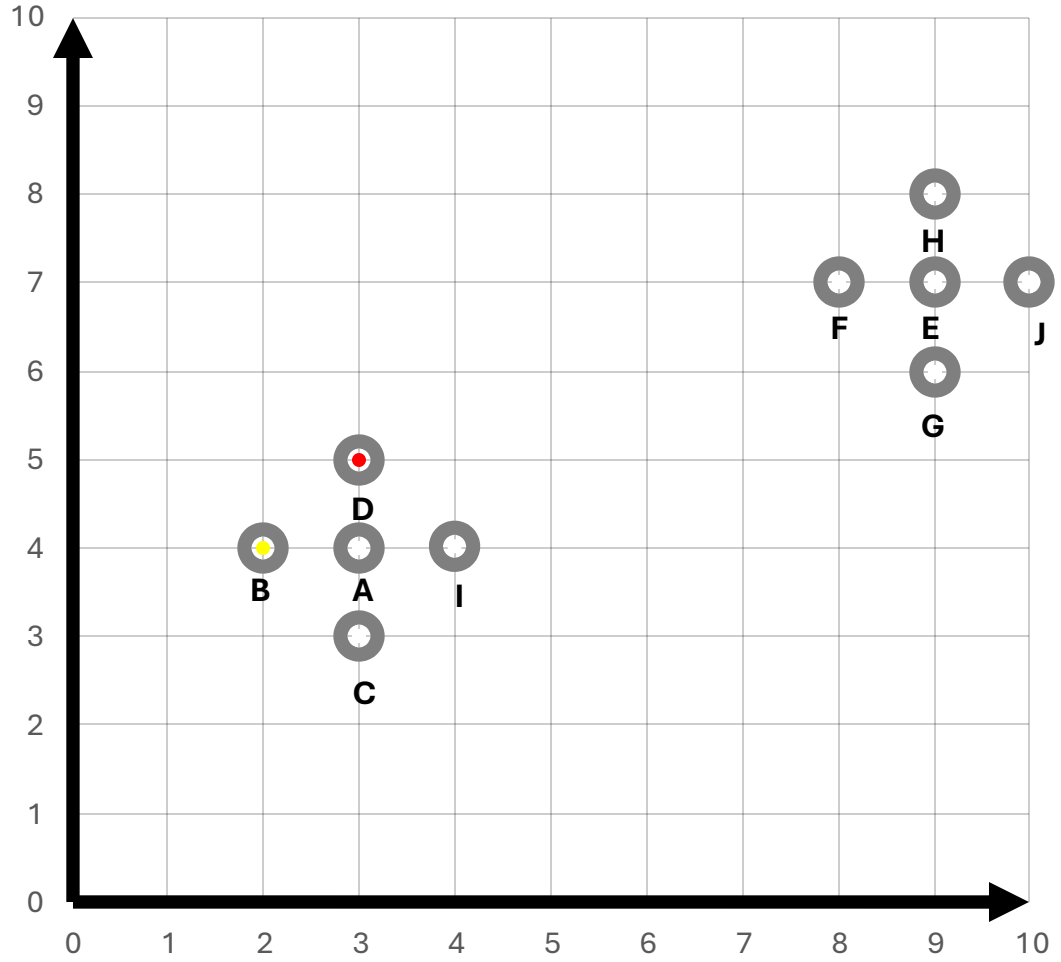
Customer	$Average\ Bet\ per\ Visit$ (x_1)	$Visit\ Per\ Month$ (x_2)
A	3	4
B	2	4
C	3	3
D	3	5
E	9	7
F	8	7
G	9	6
H	9	8
I	4	4
J	10	7

eps is the maximum distance between two samples for one to be considered as in the neighborhood of the other.

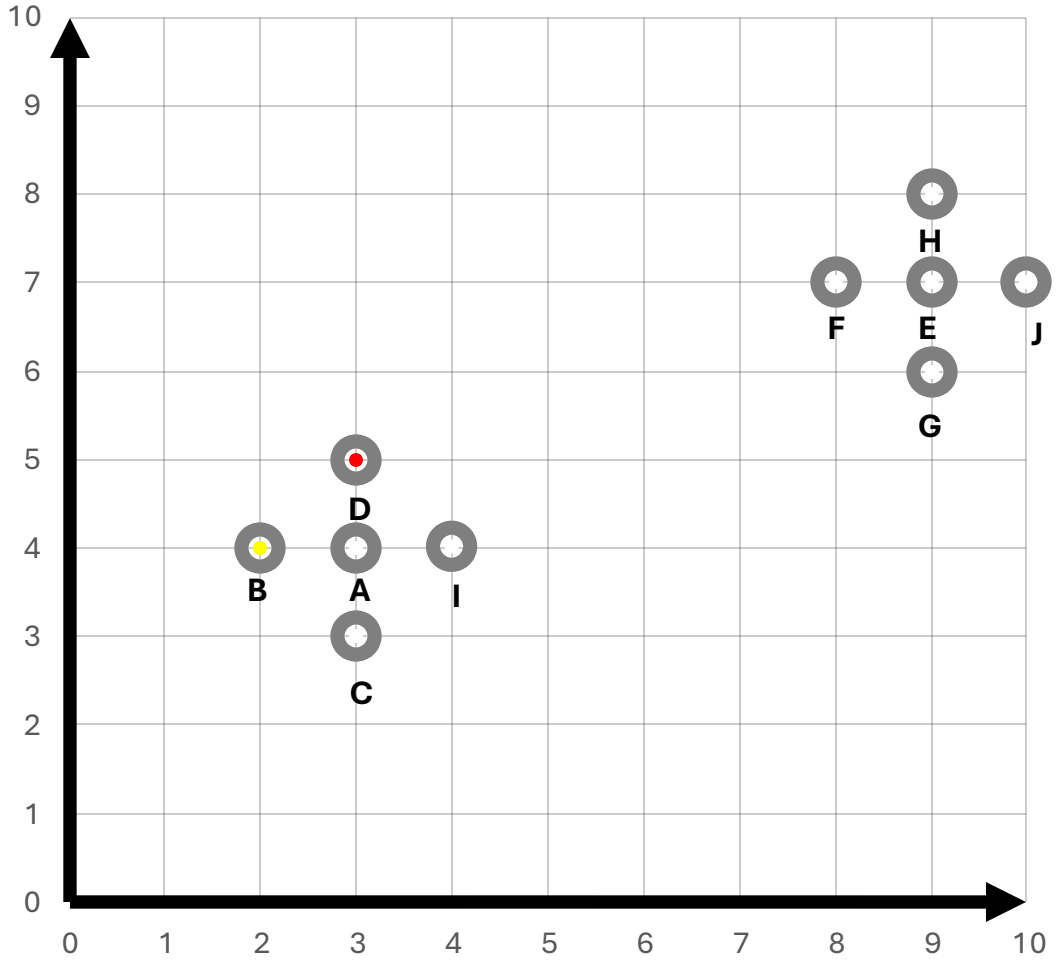


Step 1: Set **eps** and **min_samples**

min_samples is the number of samples in a neighborhood for a point to be considered as a core point.

This **includes the point itself**.

If **min_samples** is set to a higher value, DBSCAN will find denser clusters, whereas if it is set to a lower value, the found clusters will be more sparse.



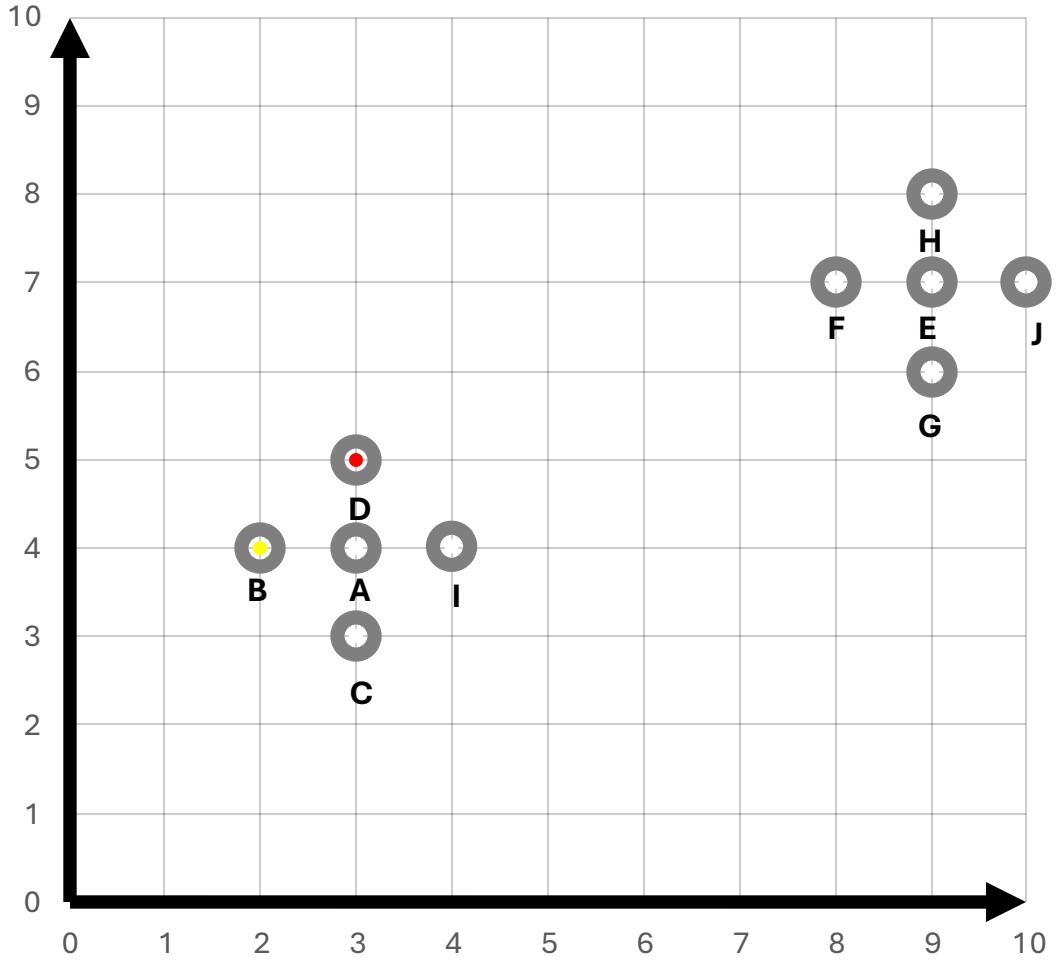
Step 1: Set **eps** and **min_samples**

For this example, we will set

eps = 1

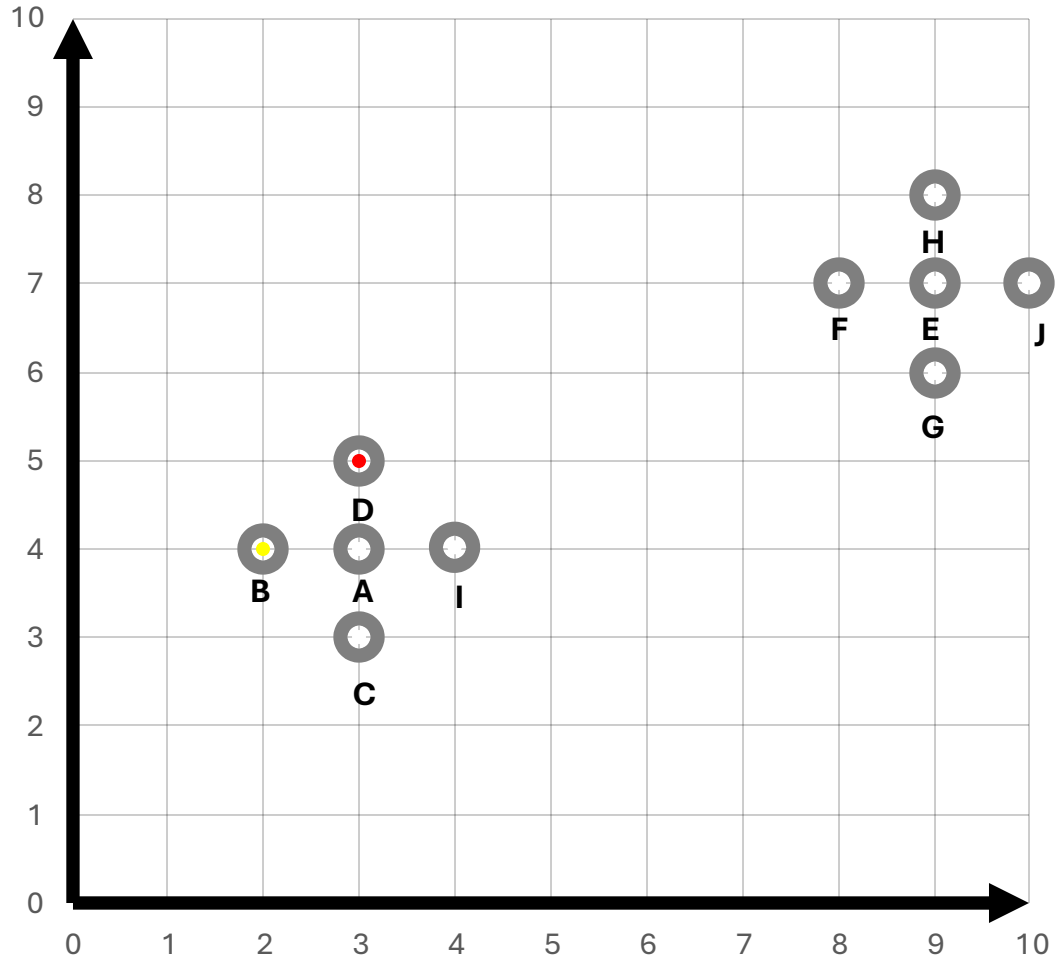
and

min_samples = 4



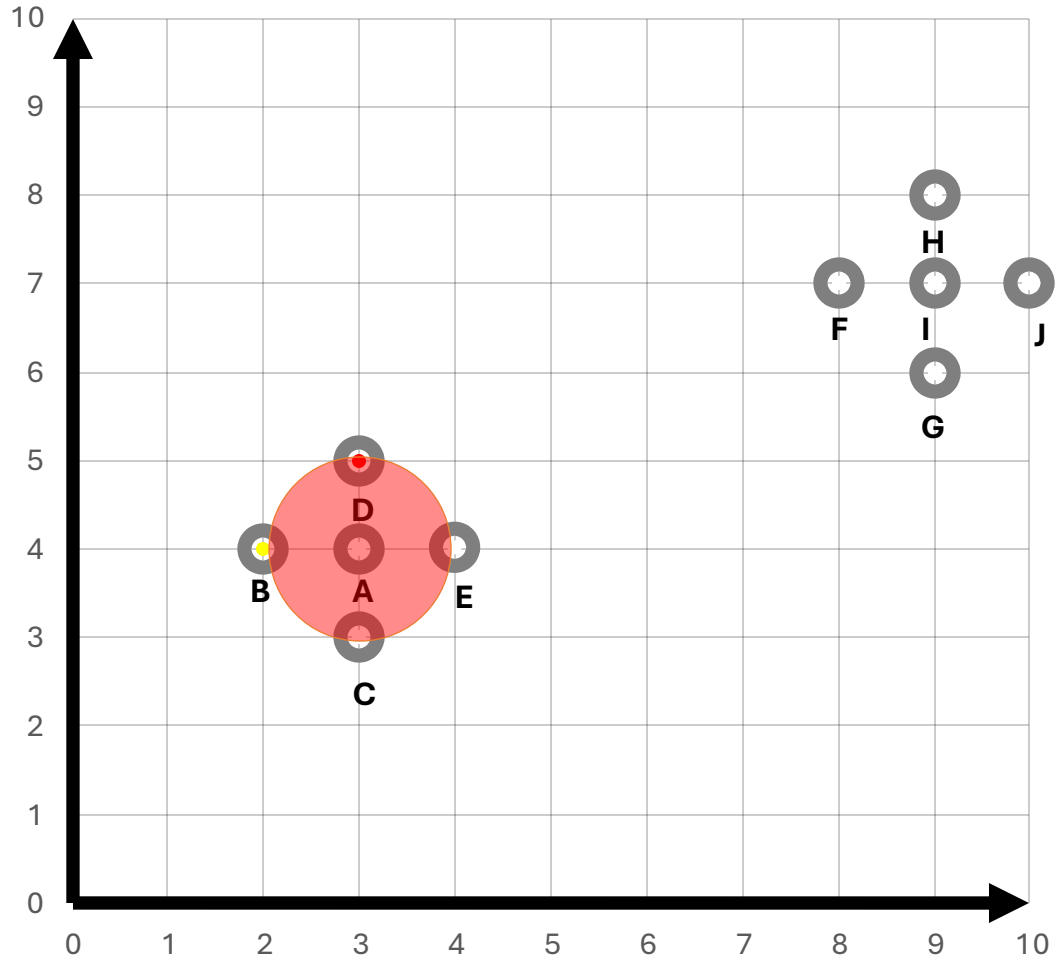
Step 1: Set **eps** and **min_samples**

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (9,7)	F (8,7)	G (9,6)	H (9,8)	I (4,4)	J (10,7)
A (3,4)	0	1	1	1	6.7	9.8	6.3	7.2	1	7.6
B (2,4)	1	0	1.4	1.4	7.6	6.7	7.2	8	2	8.5
C (3,3)	1	1.4	0	2	7.2	6.4	6.7	7.8	1.4	8
D (3,5)	1	1.4	2	0	6.3	5.3	6	6.7	1.4	7.2
E (9,7)	6.7	7.6	7.2	6.3	0	1	1	1	5.8	1
F (8,7)	9.8	6.7	6.4	5.3	1	0	1.4	1.4	5	2
G (9,6)	6.3	7.2	6.7	6	1	1.4	0	2	5.3	1.4
H (9,8)	7.2	8	7.8	6.7	1	1.4	2	0	6.4	1.4
I (4,4)	1	2	1.4	1.4	5.8	5	5.3	6.4	0	6.7
J (10,7)	7.6	8.5	8	7.2	1	2	1.4	1.4	6.7	0



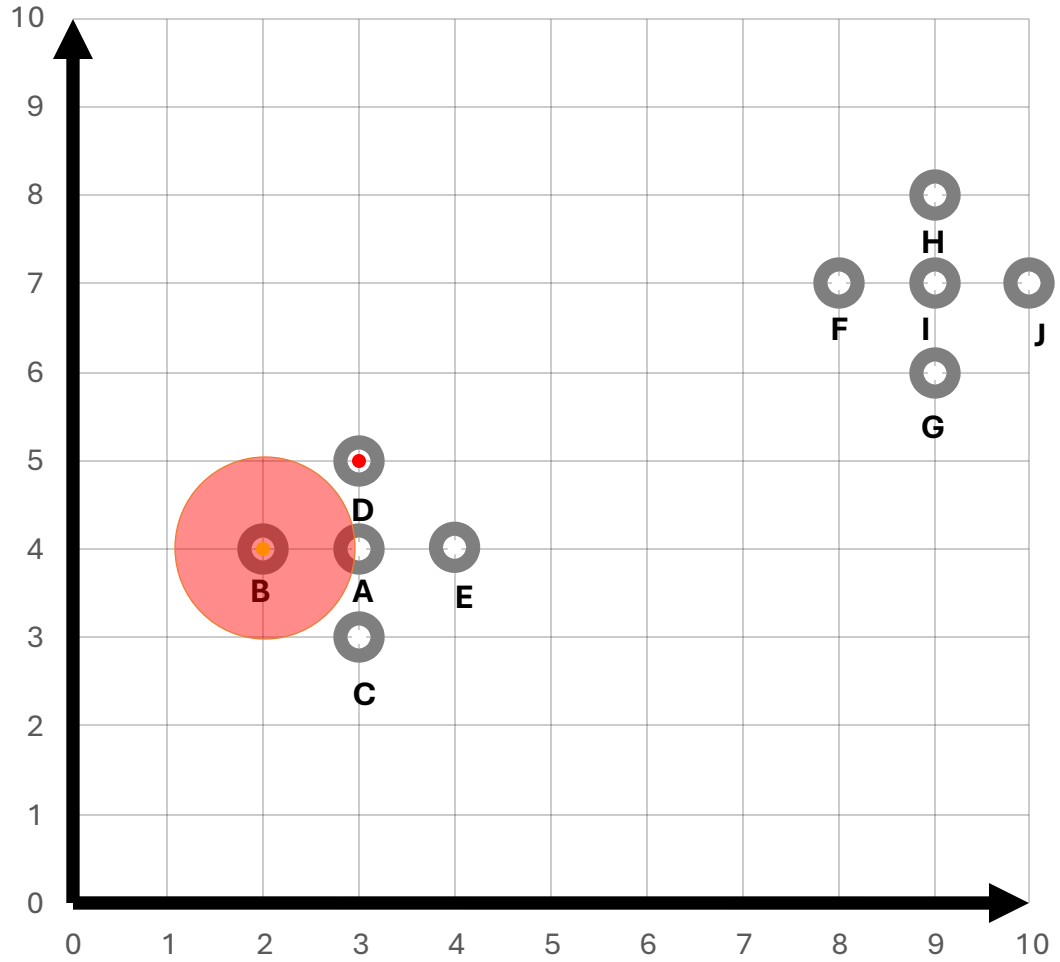
Step 2: Calculate the distance between every pair of data points using a distance metric like **Euclidean distance**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



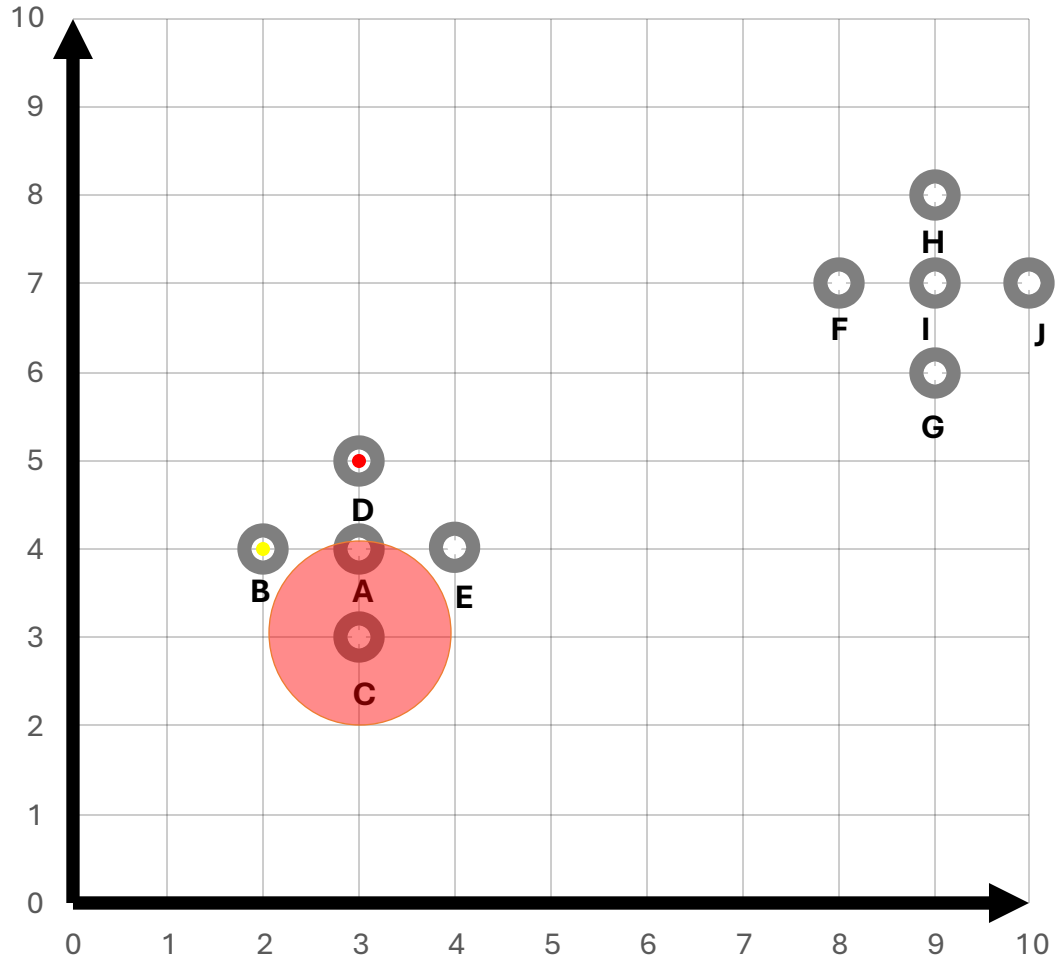
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



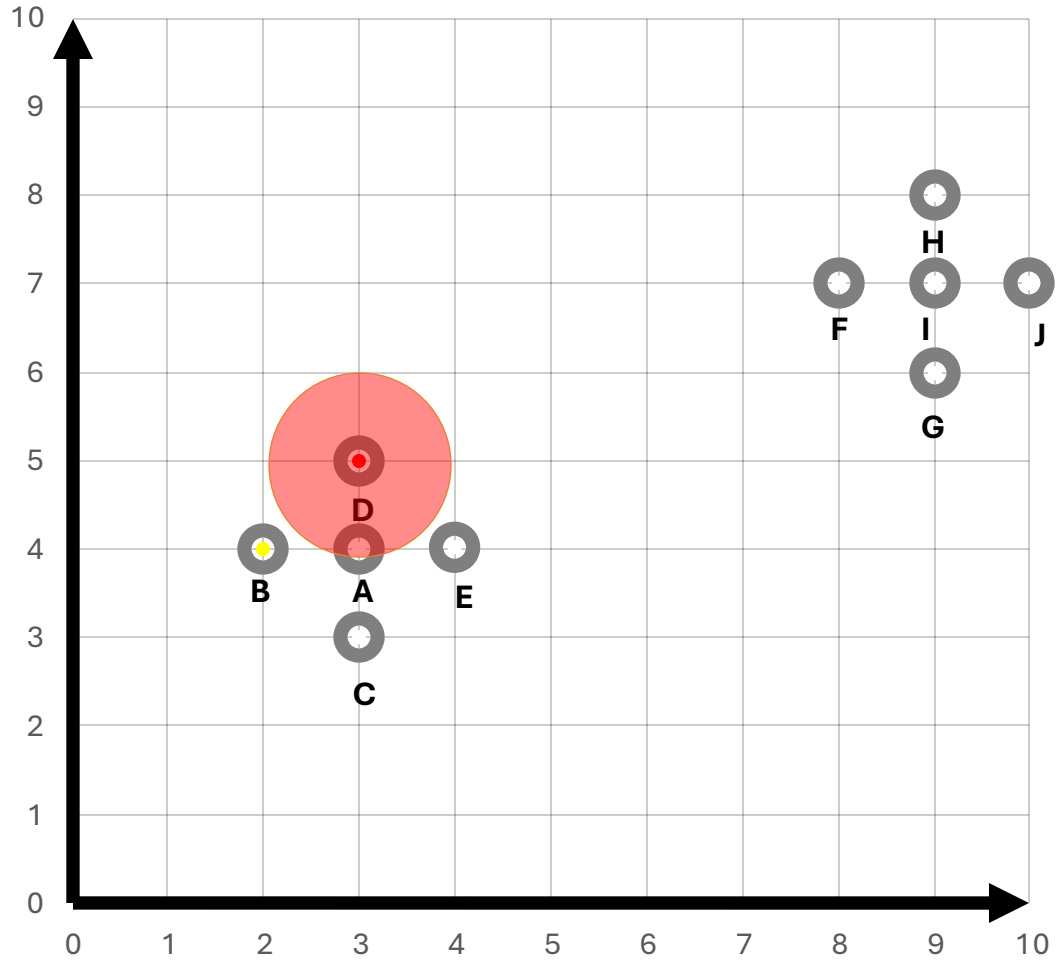
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



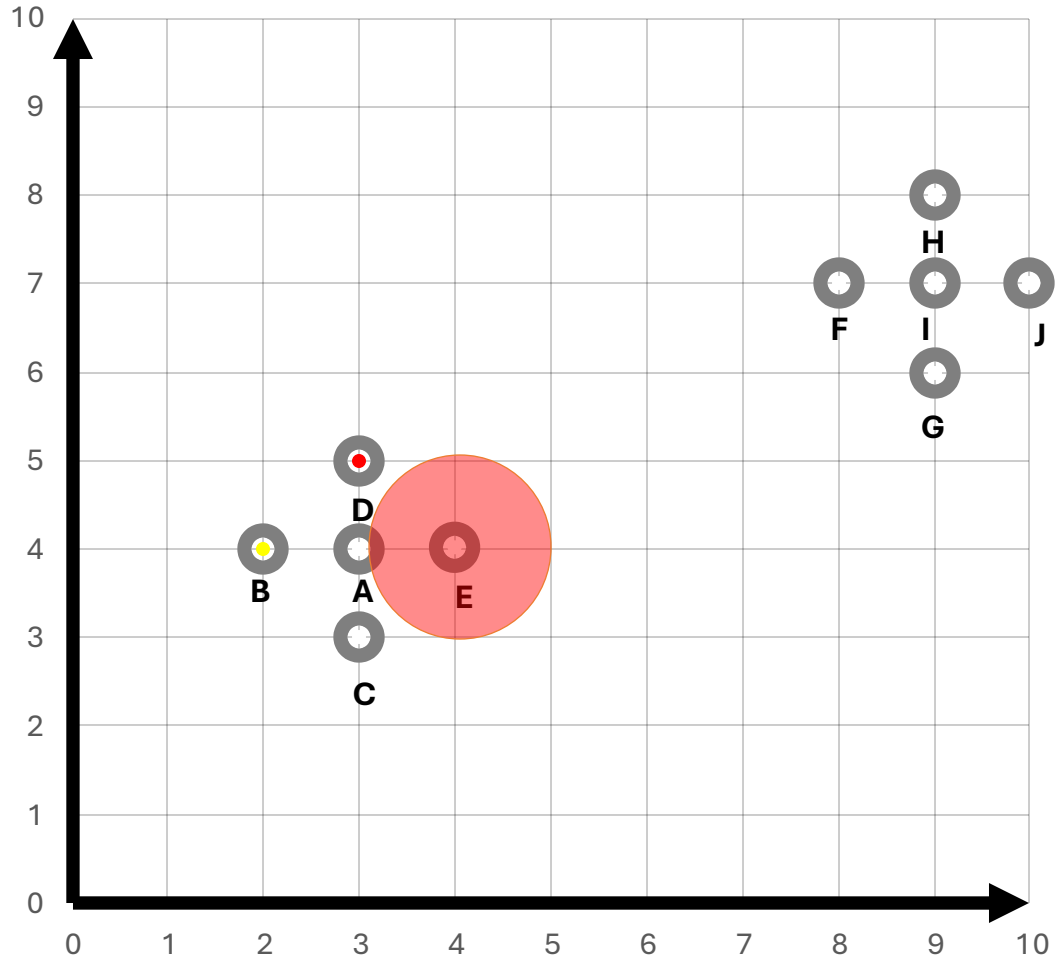
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



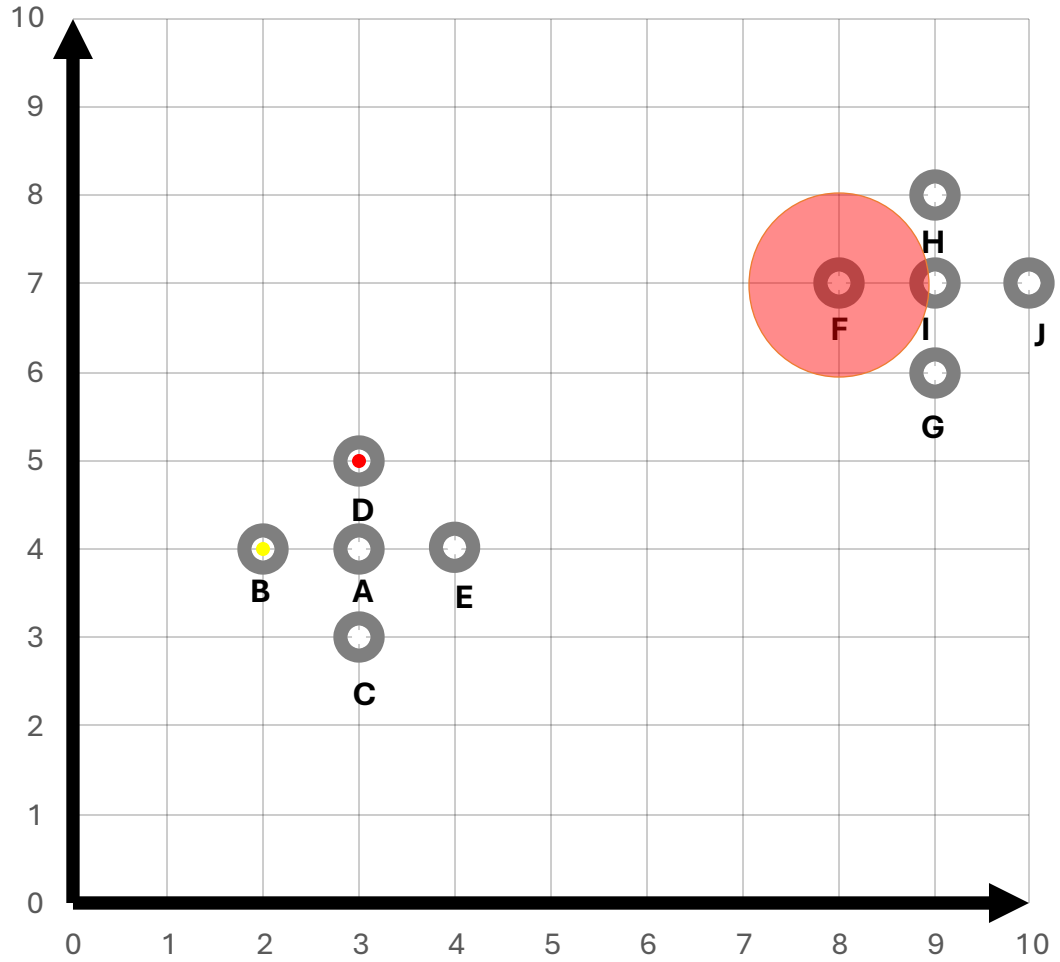
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



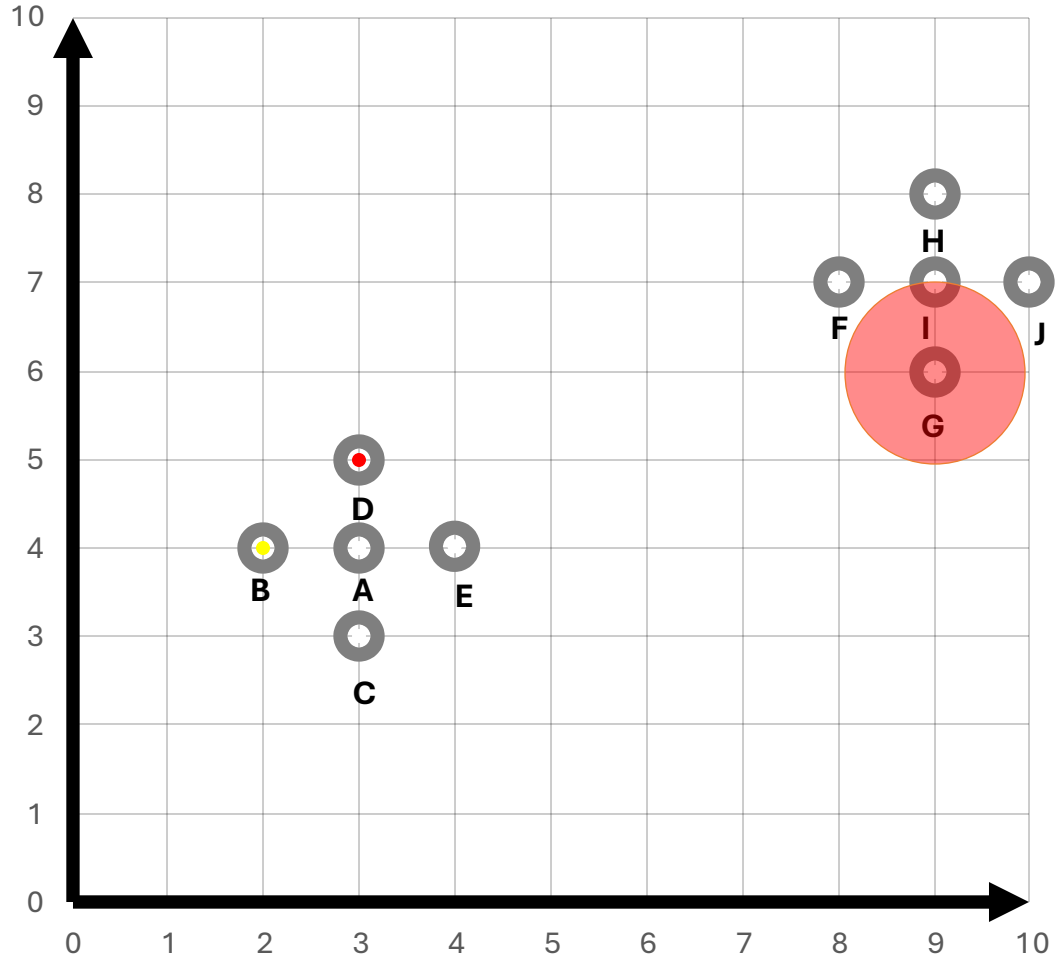
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



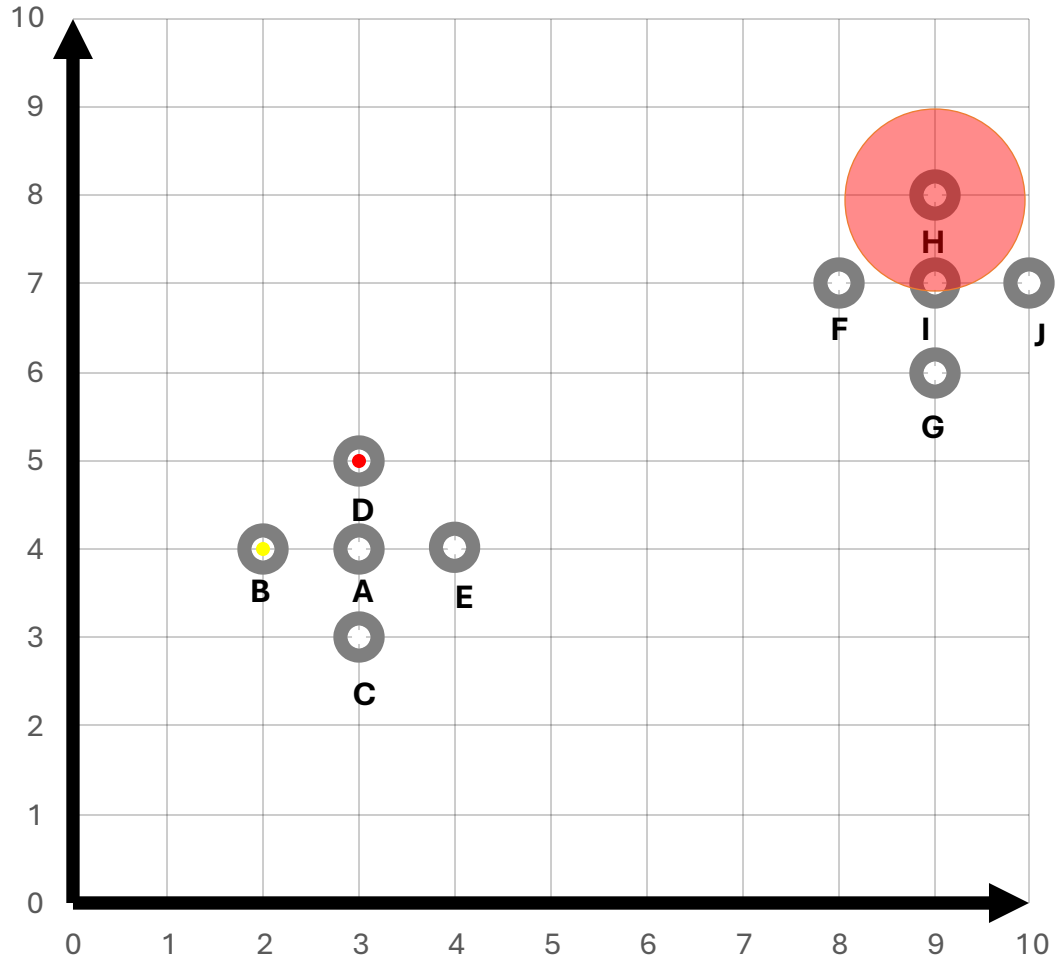
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



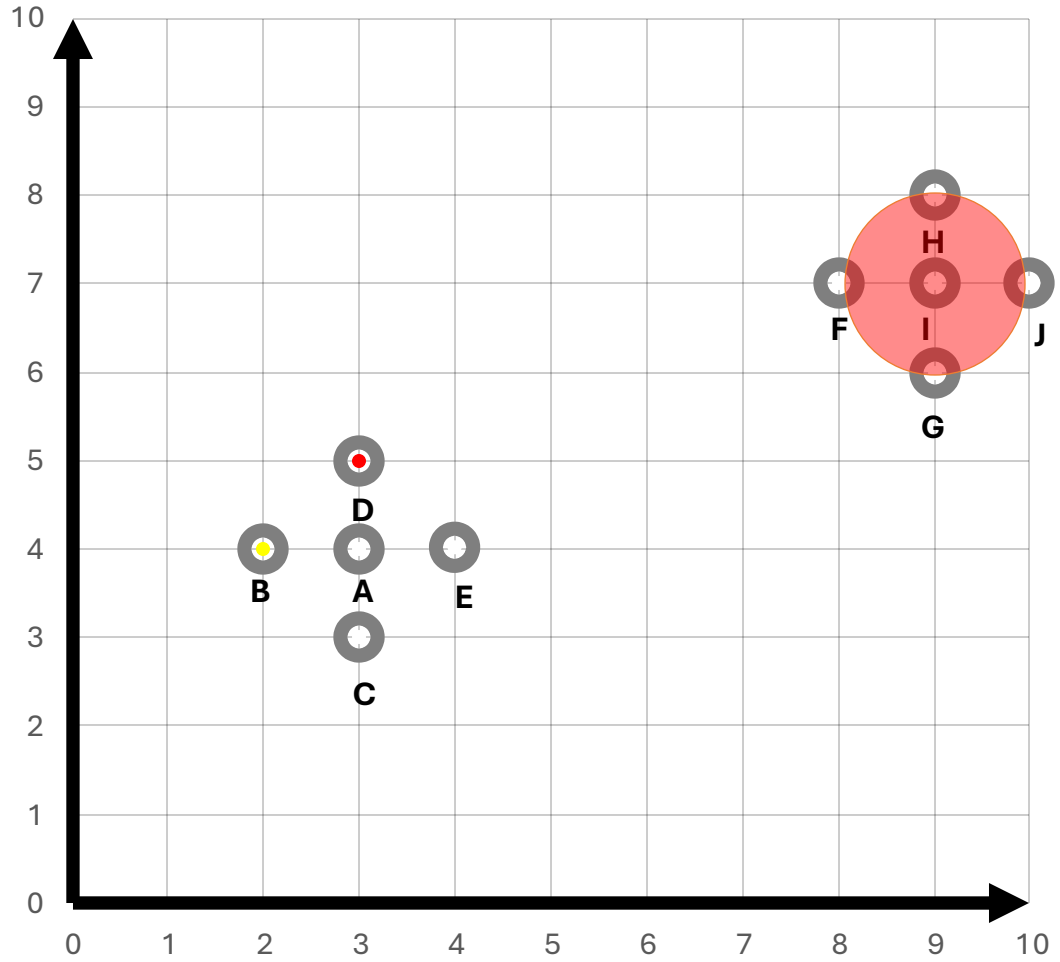
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



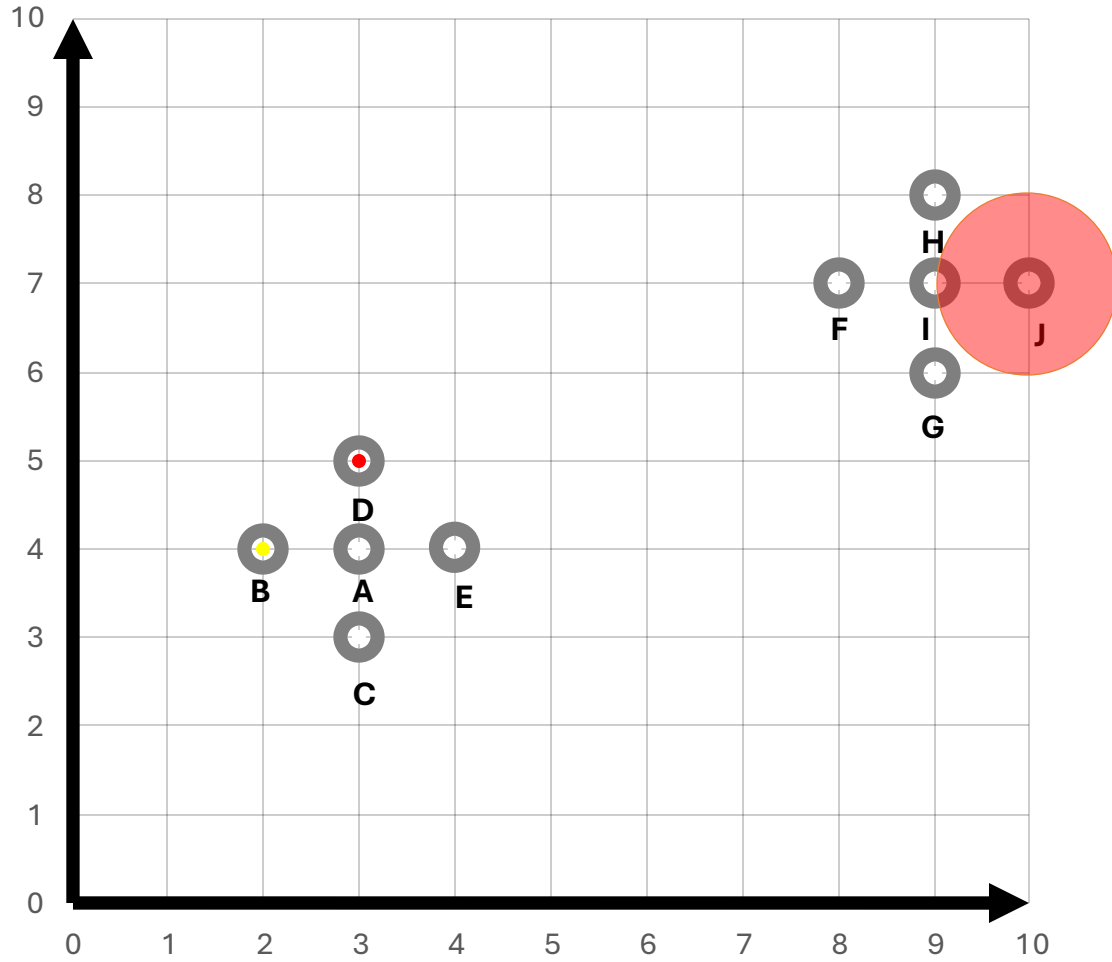
Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



Step 3: For each point, find its neighbours within the **eps/radius**.

	A (3,4)	B (2,4)	C (3,3)	D (3,5)	E (4,4)	F (8,7)	G (9,6)	H (9,8)	I (9,7)	J (10,7)
A (3,4)	0	1	1	1	1	5.8	6.3	7.2	6.7	7.6
B (2,4)	1	0	1.4	1.4	2	6.7	7.2	8	7.6	8.5
C (3,3)	1	1.4	0	2	1.4	6.4	6.7	7.8	7.2	8
D (3,5)	1	1.4	2	0	1.4	5.3	6	6.7	6.3	7.2
E (4,4)	1	2	1.4	1.4	0	5	5.3	6.4	5.8	6.7
F (8,7)	5.8	6.7	6.4	5.3	5	0	1.4	1.4	1	2
G (9,6)	6.3	7.2	6.7	6	5.3	1.4	0	2	1	1.4
H (9,8)	7.2	8	7.8	6.7	6.4	1.4	2	0	1	1.4
I (9,7)	6.7	7.6	7.2	6.3	5.8	1	1	1	0	1
J (10,7)	7.6	8.5	8	7.2	6.7	2	1.4	1.4	1	0



Step 3: For each point, find its neighbours within the **eps/radius**.

Customer	*Neighbours within eps*
A	B,C,D,E
B	A
C	A
D	A
E	I
F	I
G	I
H	I
I	F,G,H,J
J	I

Step 3: For each point, find its neighbours within the **eps/radius**.

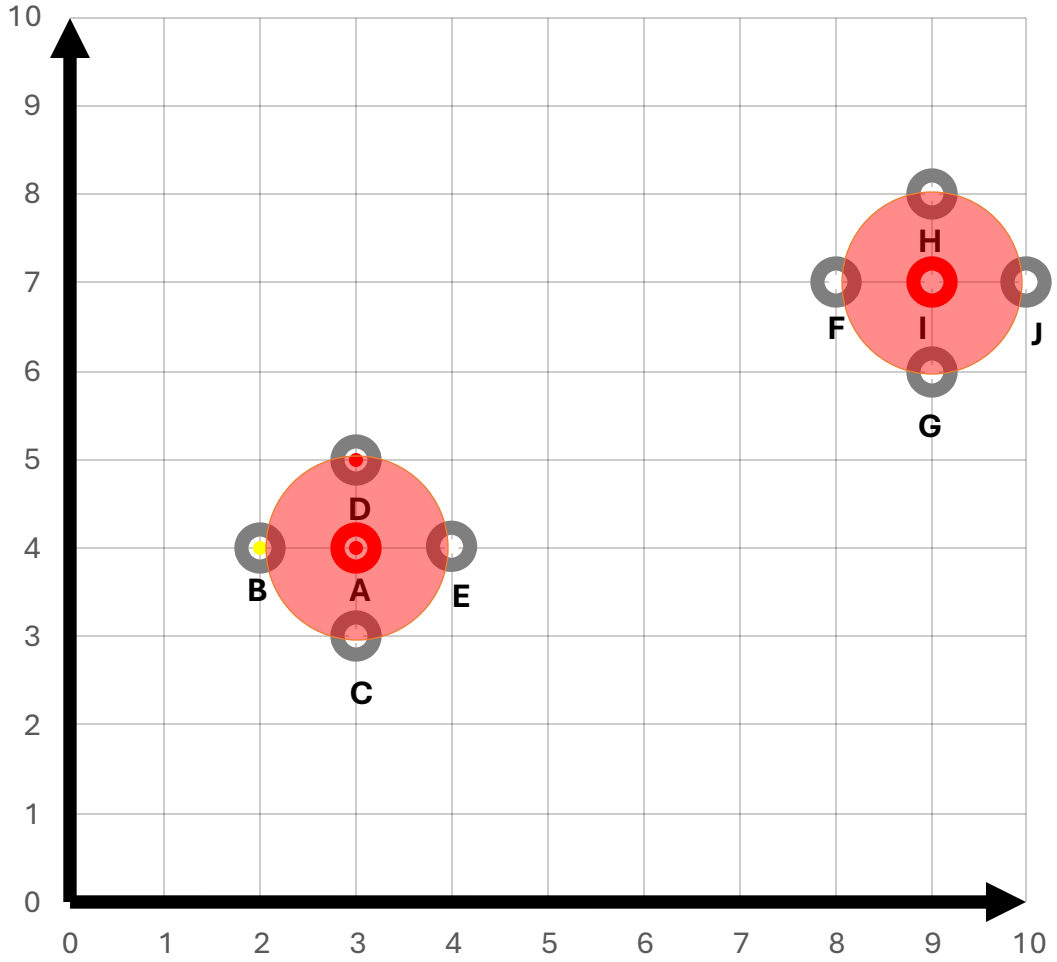
Customer	*Neighbours within eps*	*Number of neighbours*
A	B,C,D,E	4
B	A	1
C	A	1
D	A	1
E	I	1
F	I	1
G	I	1
H	I	1
I	F,G,H,J	4
J	I	1

Step 4: Determine the **core points**. To be a core point, its number of neighbours must be greater than or equal to the **min_samples**

Customer	*Neighbours within eps*	*Number of neighbours*
A	B,C,D,E	4
B	A	1
C	A	1
D	A	1
E	I	1
F	I	1
G	I	1
H	I	1
I	F,G,H,J	4
J	I	1

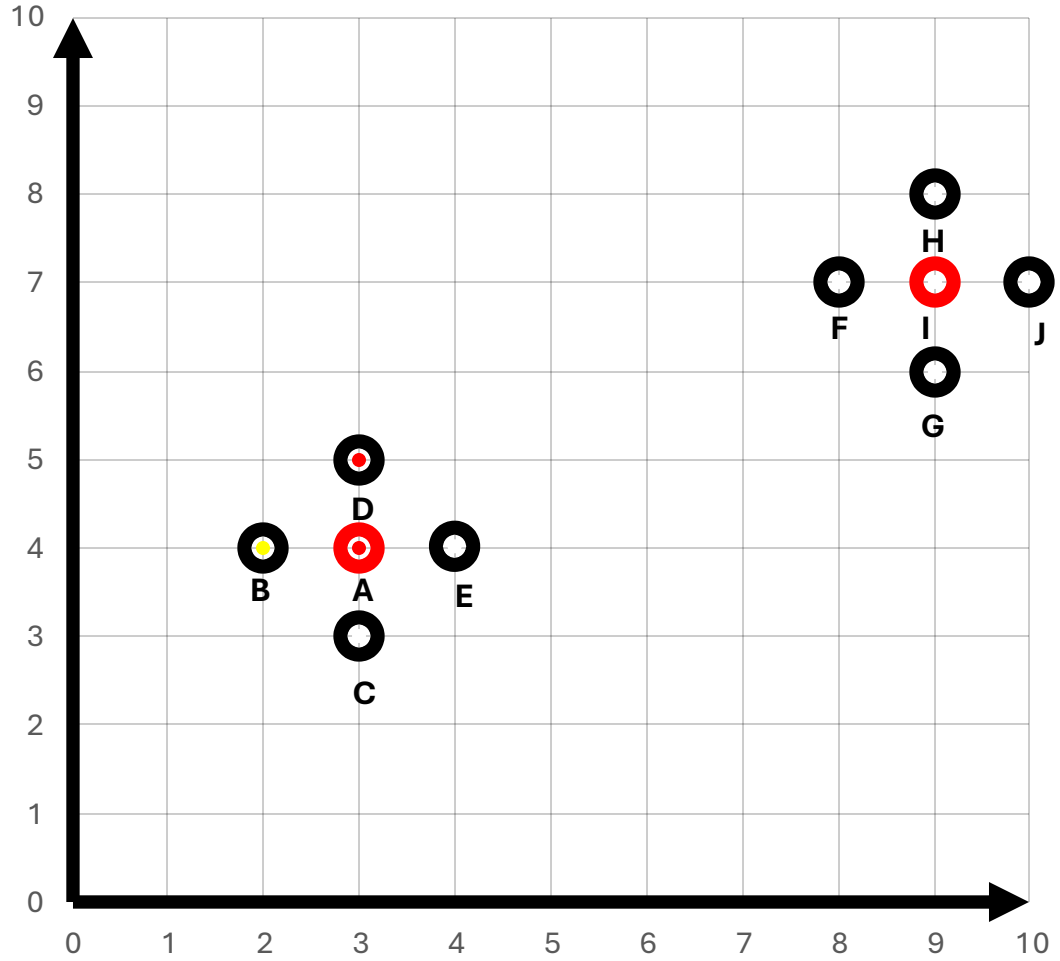
Step 4: Determine the **core points**. To be a core point, its number of neighbours must be greater than or equal to the **min_samples**

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type
A	B,C,D,E	4	**Core**
B	A	1	
C	A	1	
D	A	1	
E	I	1	
F	I	1	
G	I	1	
H	I	1	
I	F,G,H,J	4	**Core**
J	I	1	



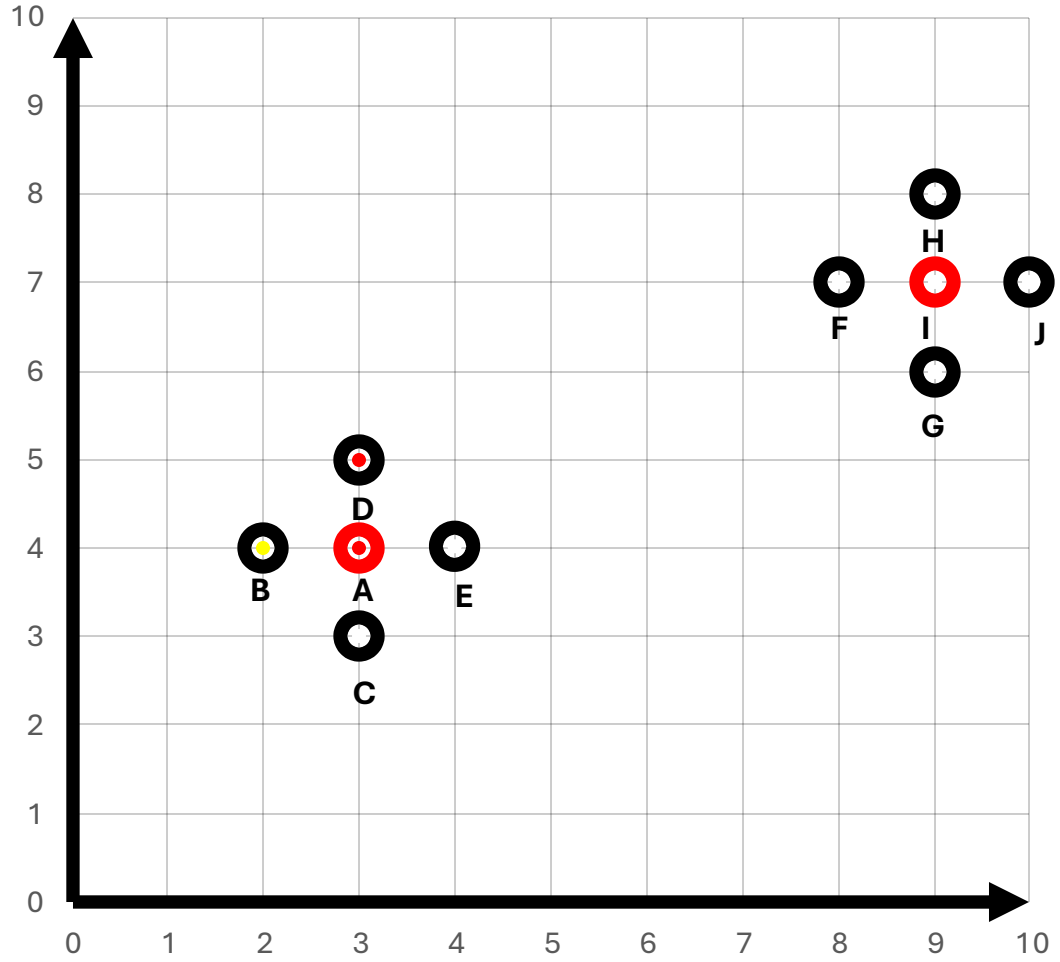
Step 4: Determine the **core points**. To be a core point, its number of neighbours must be greater than or equal to the **min_samples**

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type
A	B,C,D,E	4	**Core**
B	A	1	**Non-core**
C	A	1	**Non-core**
D	A	1	**Non-core**
E	I	1	**Non-core**
F	I	1	**Non-core**
G	I	1	**Non-core**
H	I	1	**Non-core**
I	F,G,H,J	4	**Core**
J	I	1	**Non-core**



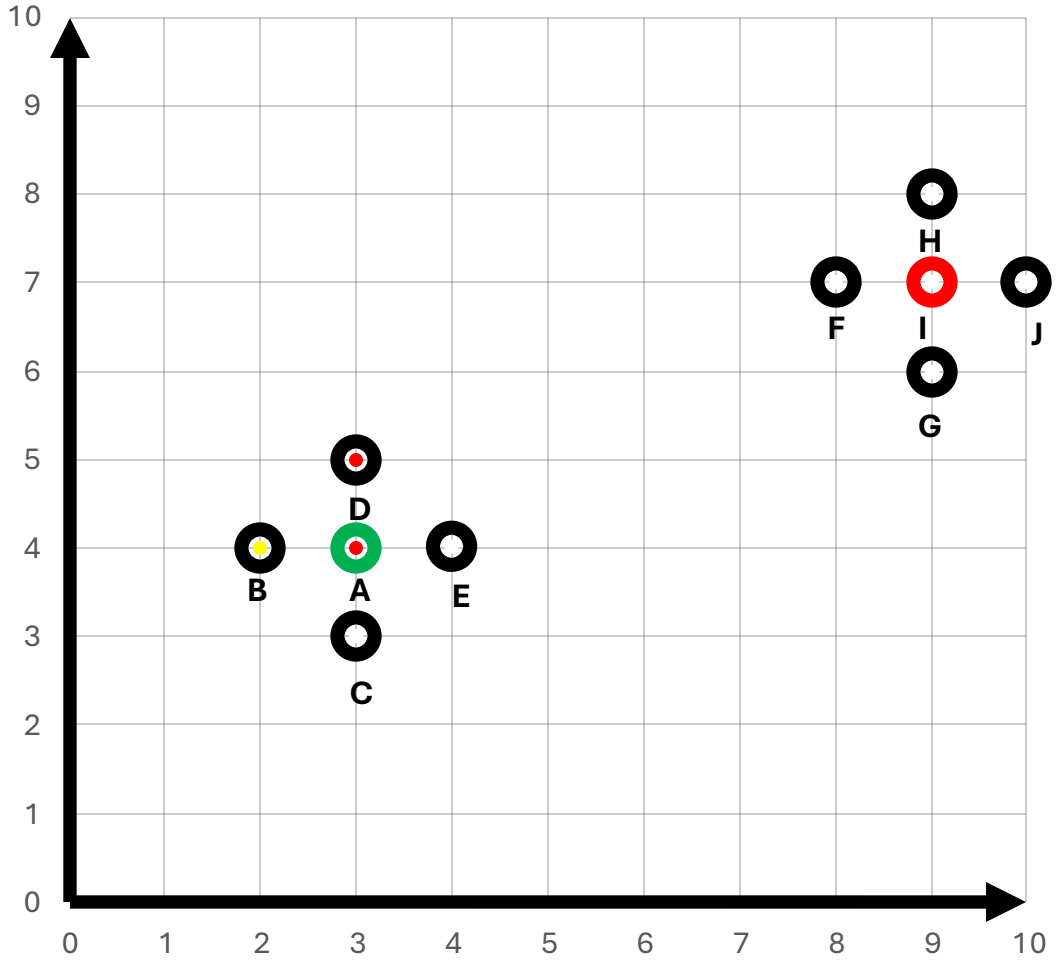
Step 5: Determine the **non-core points**. A data point will be a non-core point if its number of neighbours is less than the **min_samples.**

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	
B	A	1	**Non-core**	
C	A	1	**Non-core**	
D	A	1	**Non-core**	
E	I	1	**Non-core**	
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



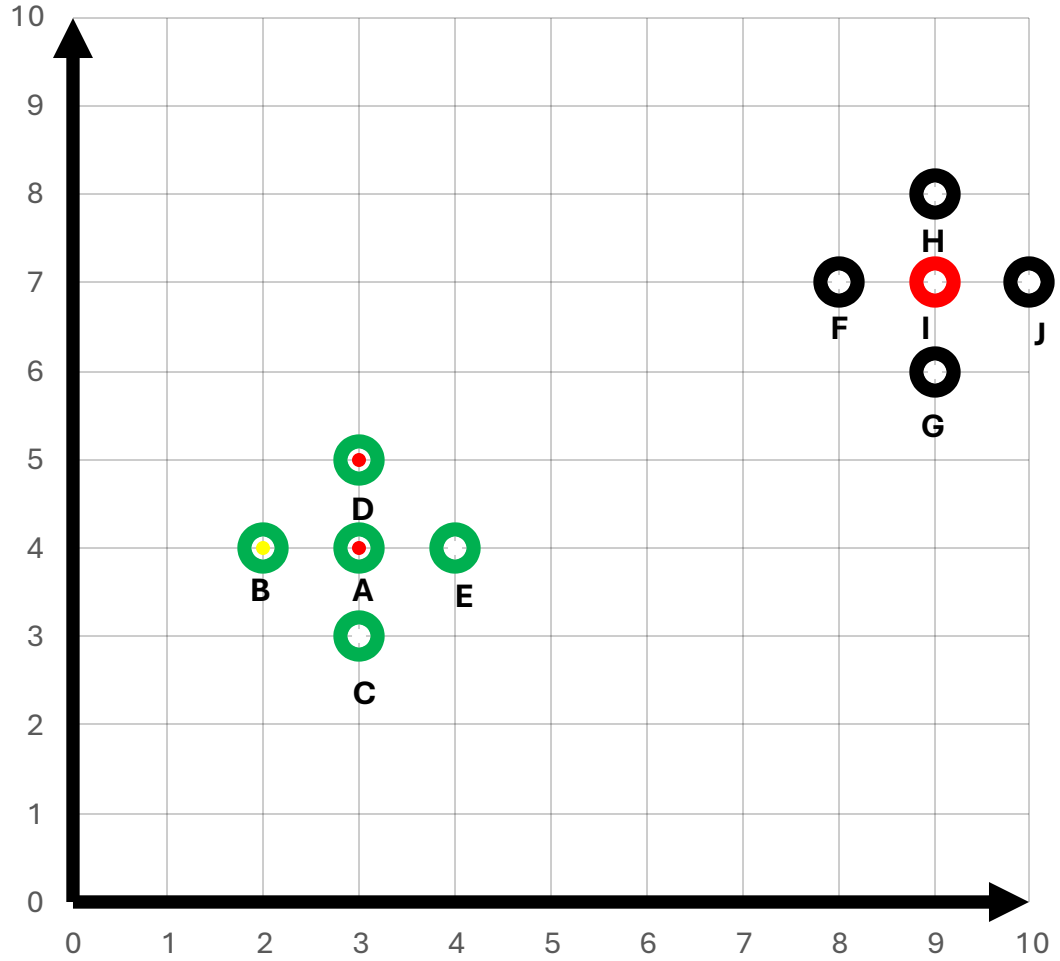
Step 6: Randomly pick a **core point** and add it to the **first cluster**. Here we first select **core point A.**

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	
C	A	1	**Non-core**	
D	A	1	**Non-core**	
E	I	1	**Non-core**	
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



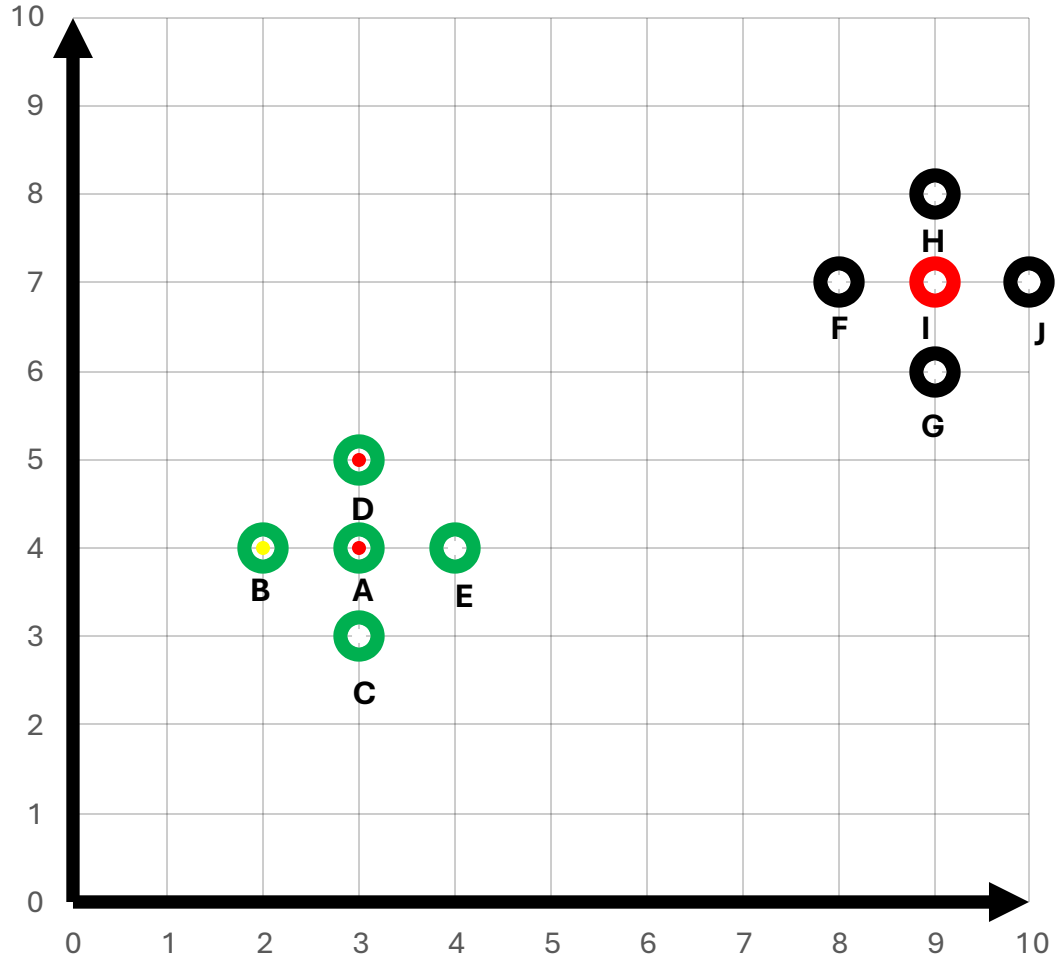
We then assign **A to** the **first cluster**.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



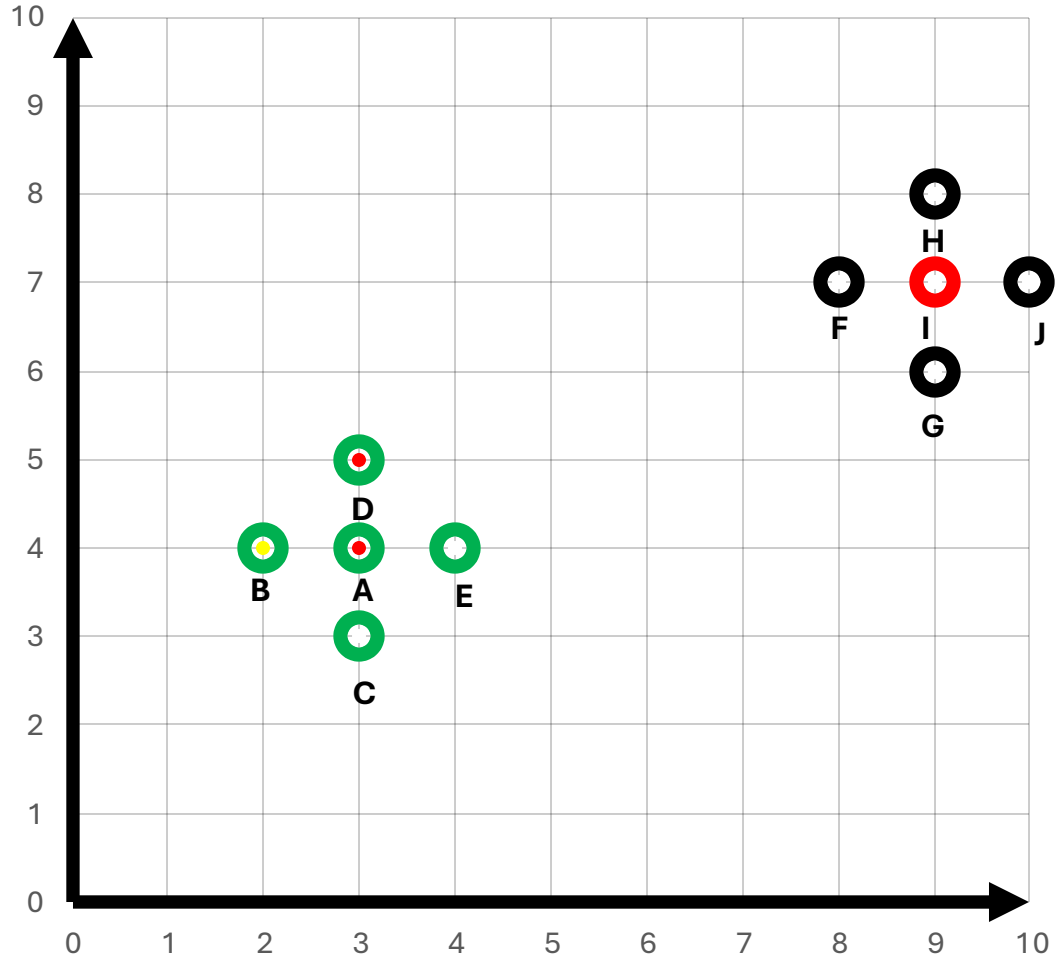
Expand the **first cluster** from **A,** we add **B,C,D,E,** to the **first cluster**

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



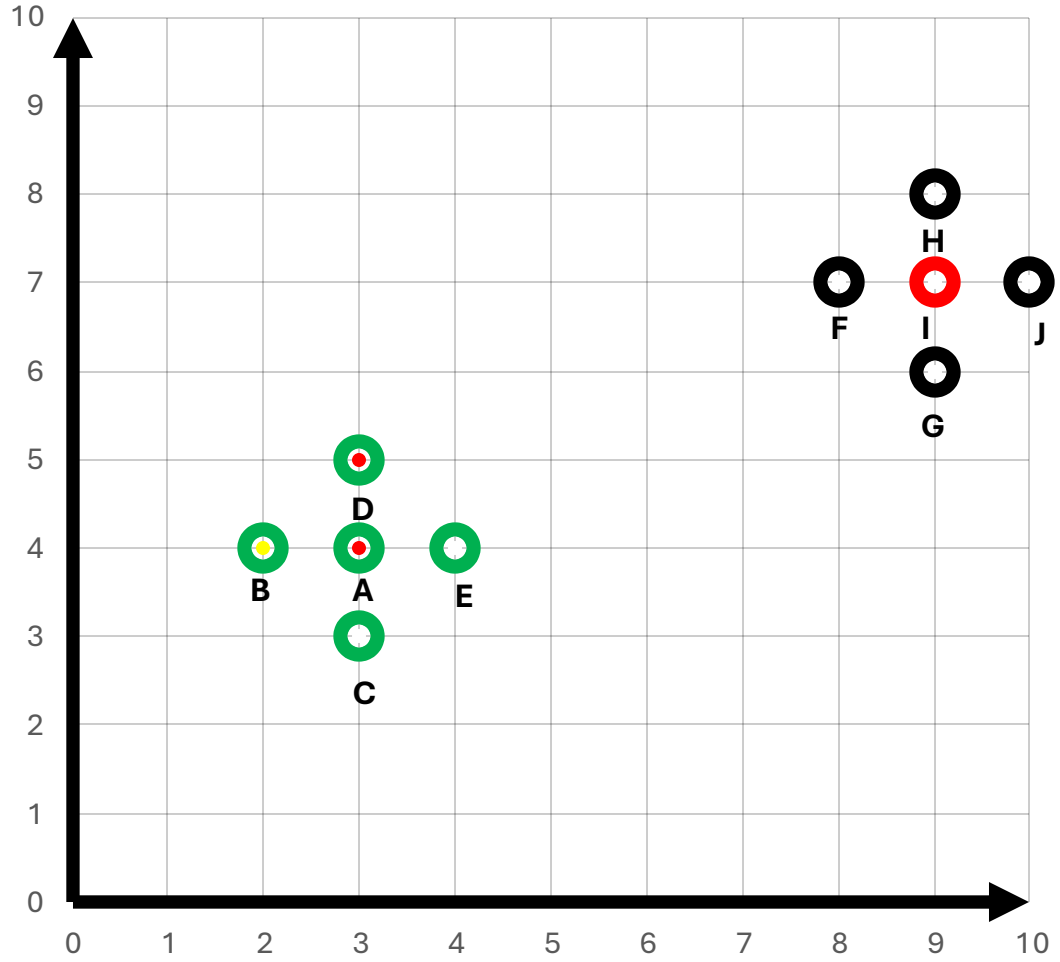
Move to point **B**, since **B** is a **non-core point,** we do not expand further

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



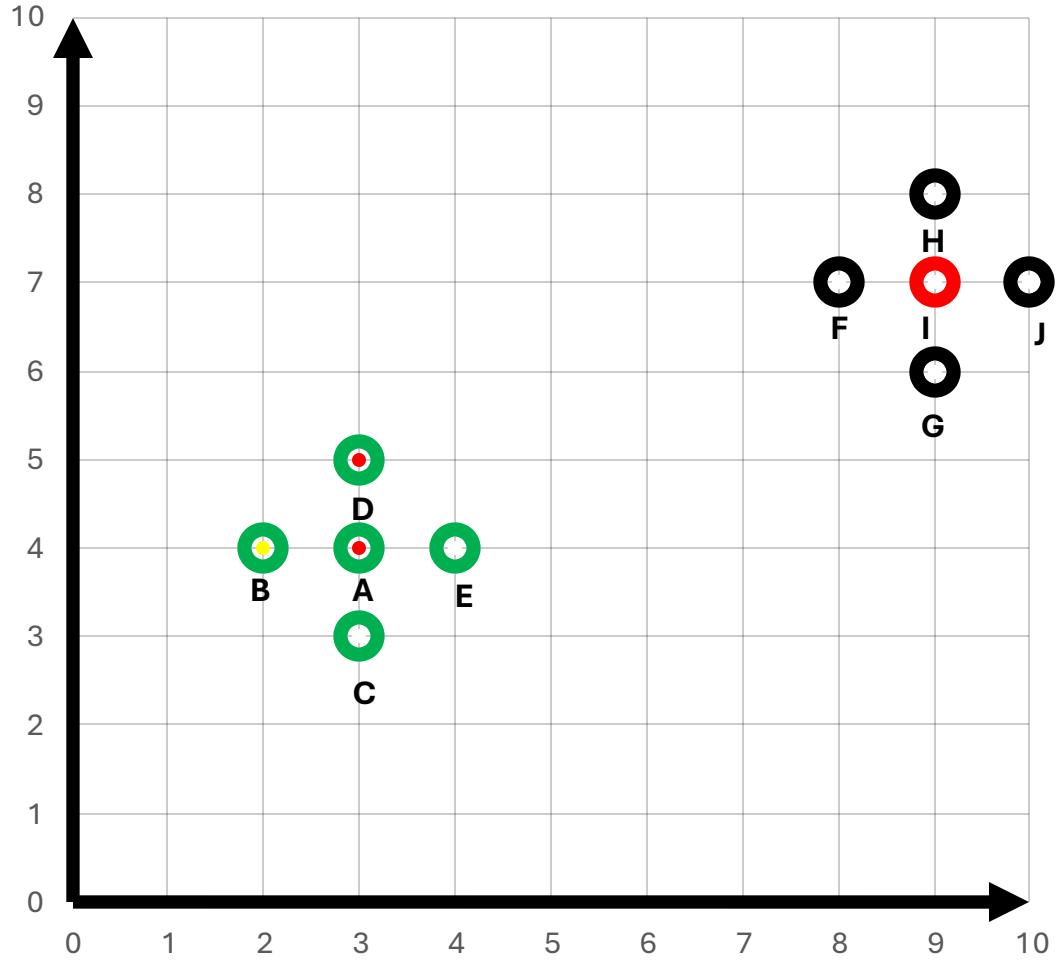
We then move to point **C**, **D** then **E**. Since they are all **non-core points**, we do not expand further.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



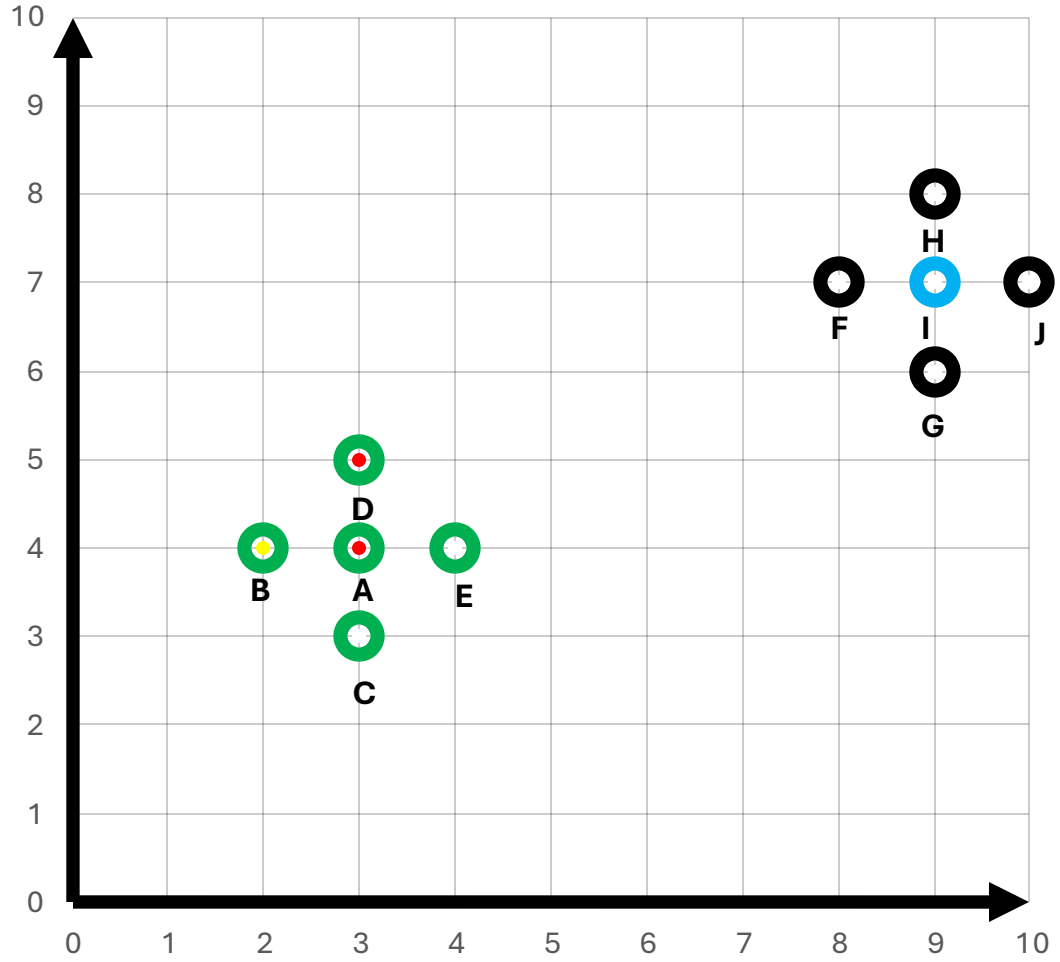
We then move to point **C**, **D** then **E**. Since they are all **non-core points**, we do not expand further.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	
J	I	1	**Non-core**	



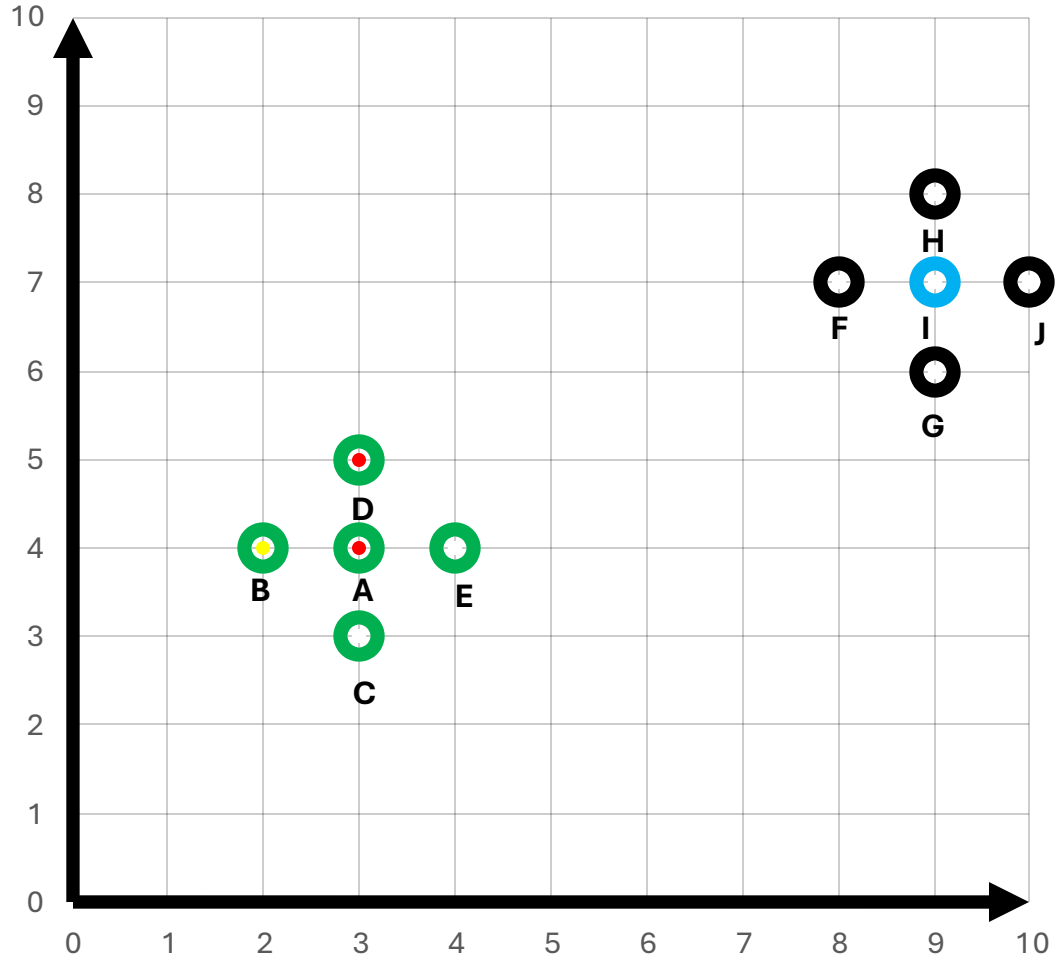
We then move to point **C**, **D** then **E**. Since they are all **non-core points**, we do not expand further.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	
G	I	1	**Non-core**	
H	I	1	**Non-core**	
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	



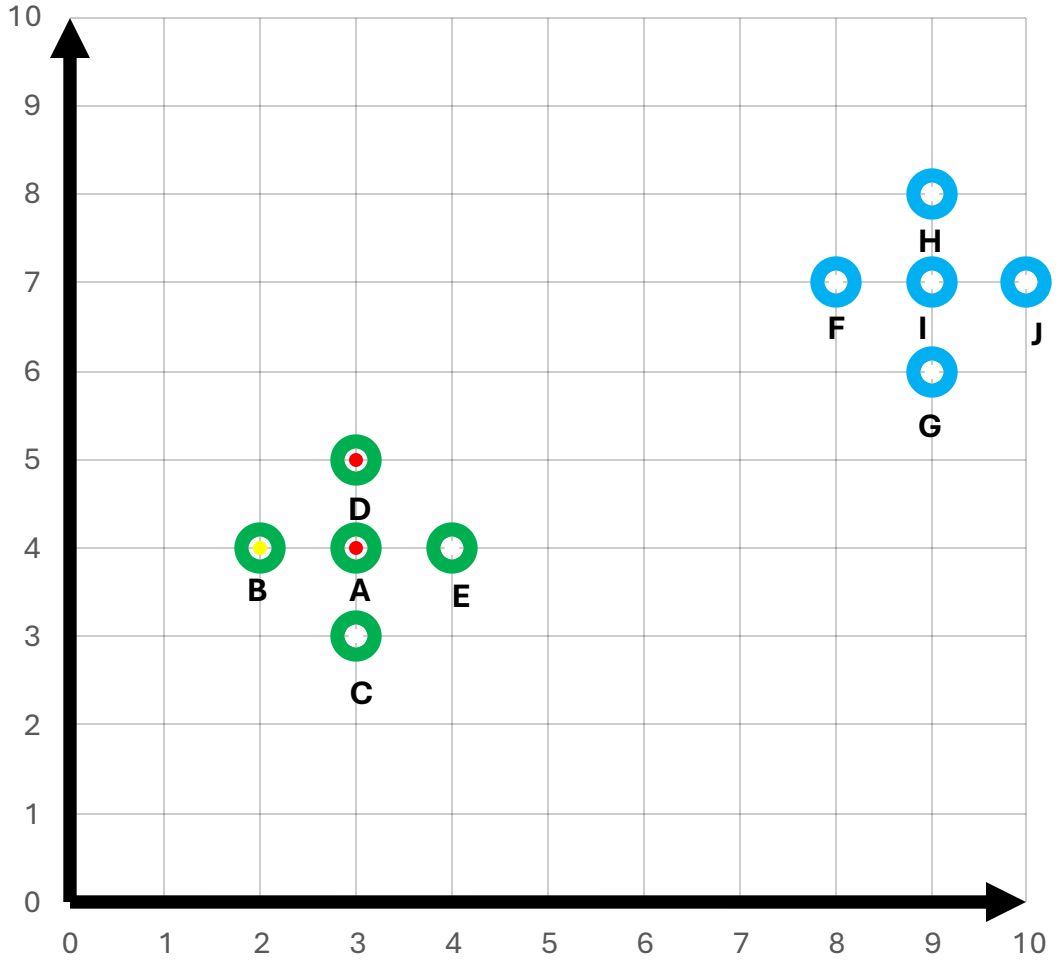
For the remaining unvisited **core points,** start a **second cluster**.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**



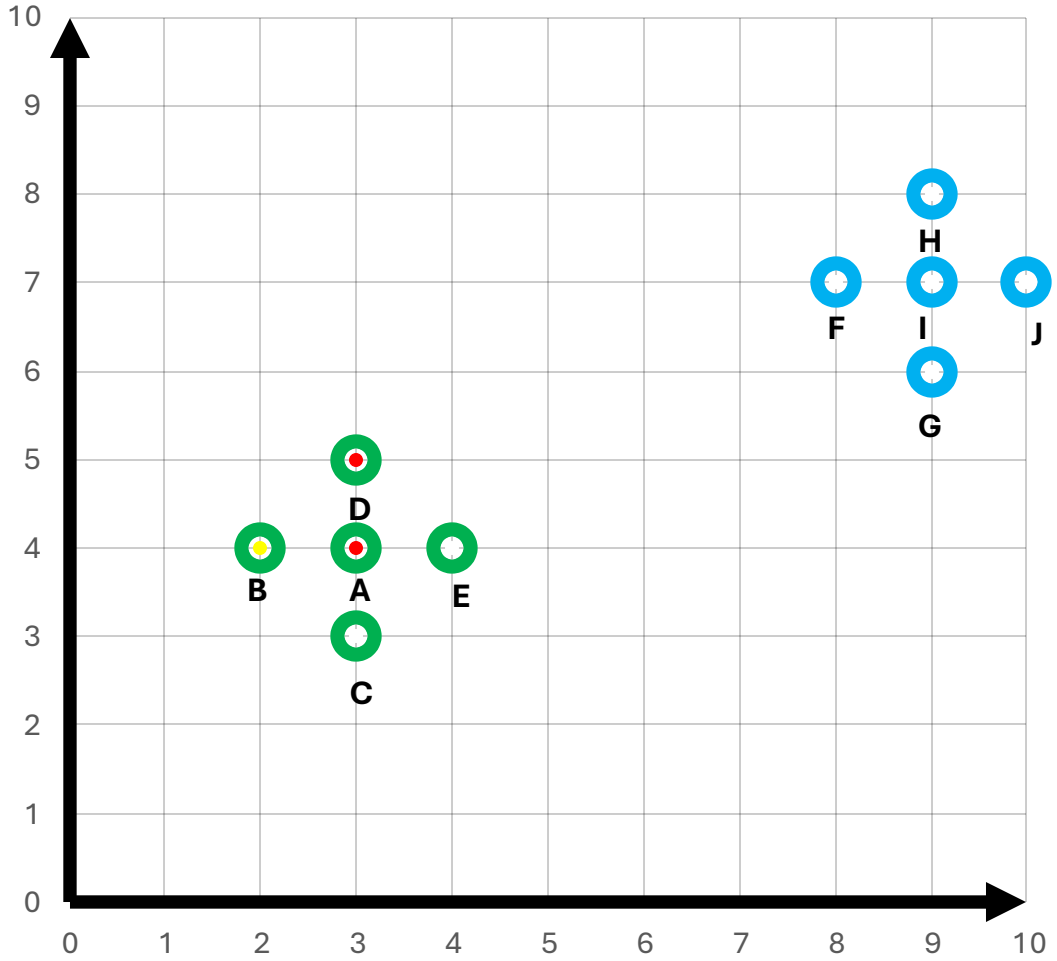
Expand the **second cluster** from **I.** Add **F,G,H,J,** to the **second cluster**

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**



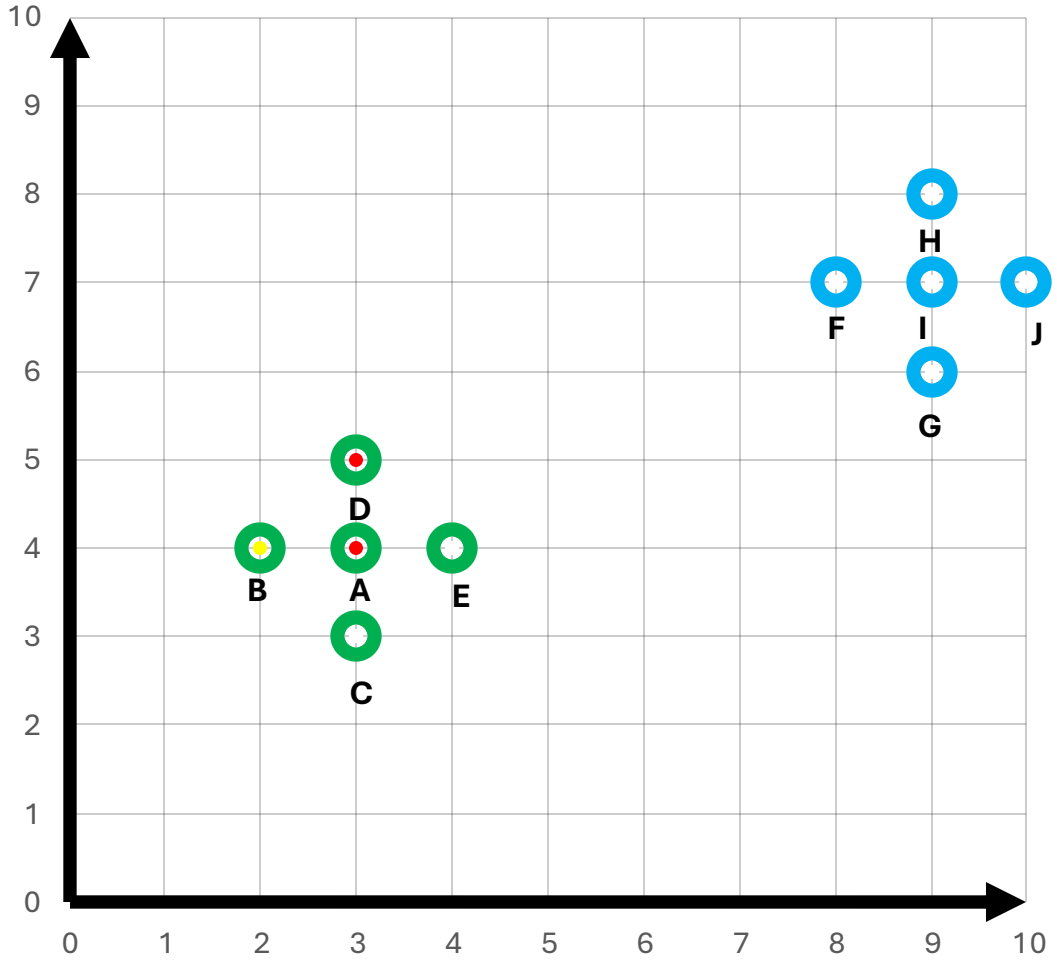
Move to point **F. Since F** is a **non-core point,** we do not expand the **second cluster** further

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**



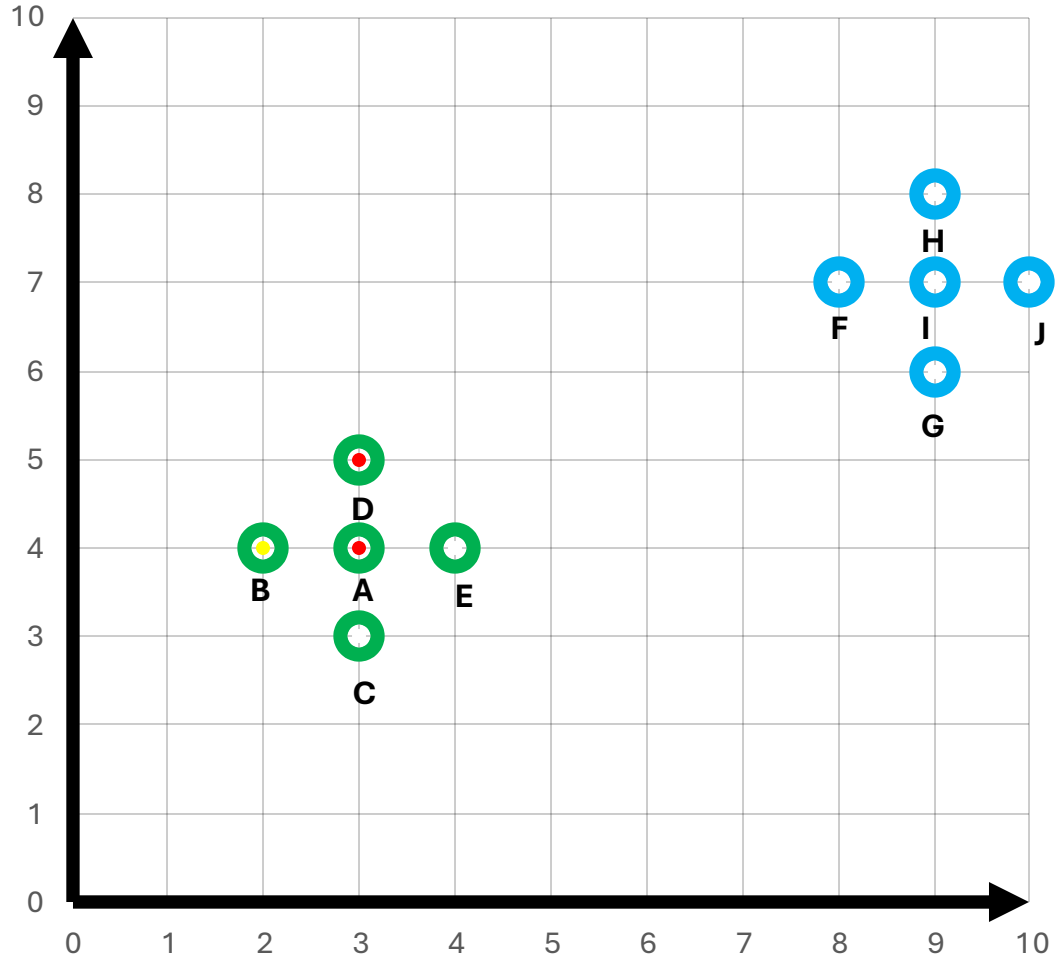
We then move to point **G**, **H** then **J**. Since they are all **non-core points**, we do not expand further.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**



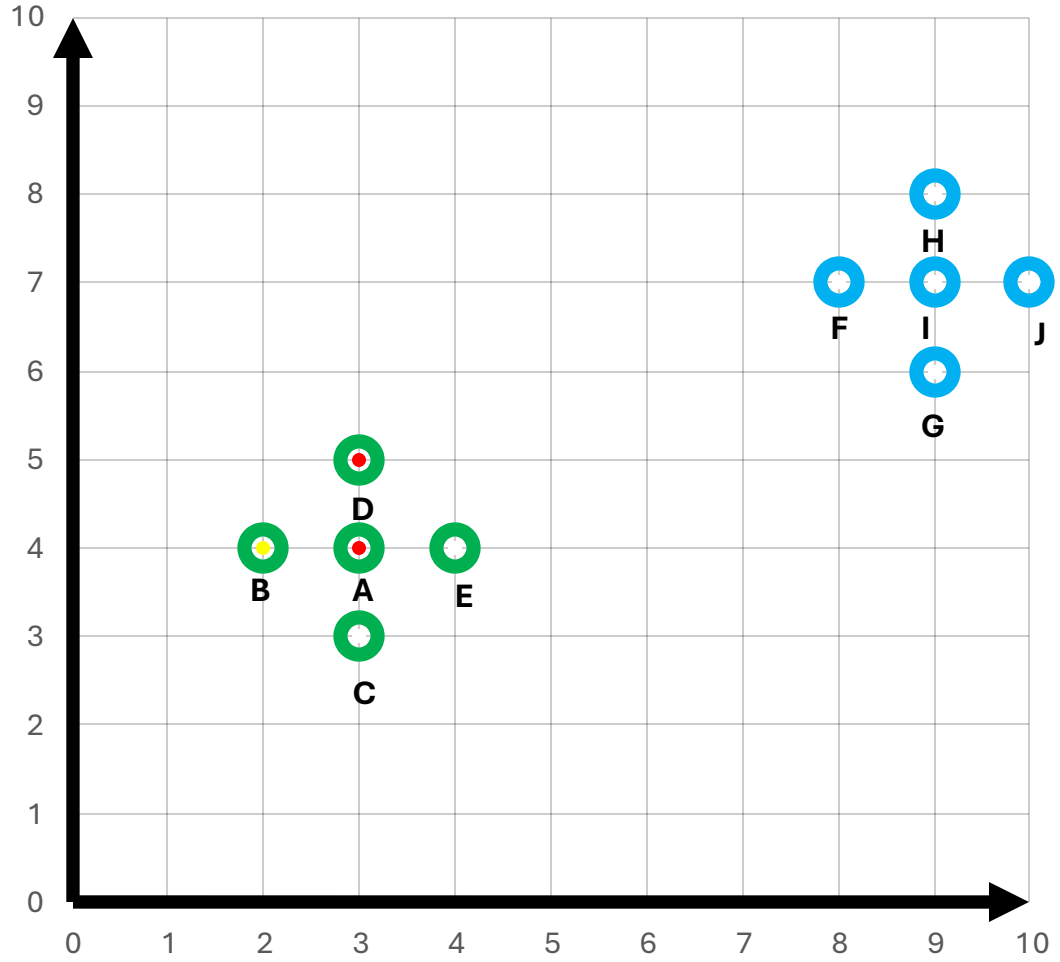
We then move to point **G**, **H** then **J**. Since they are all **non-core points**, we do not expand further.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**

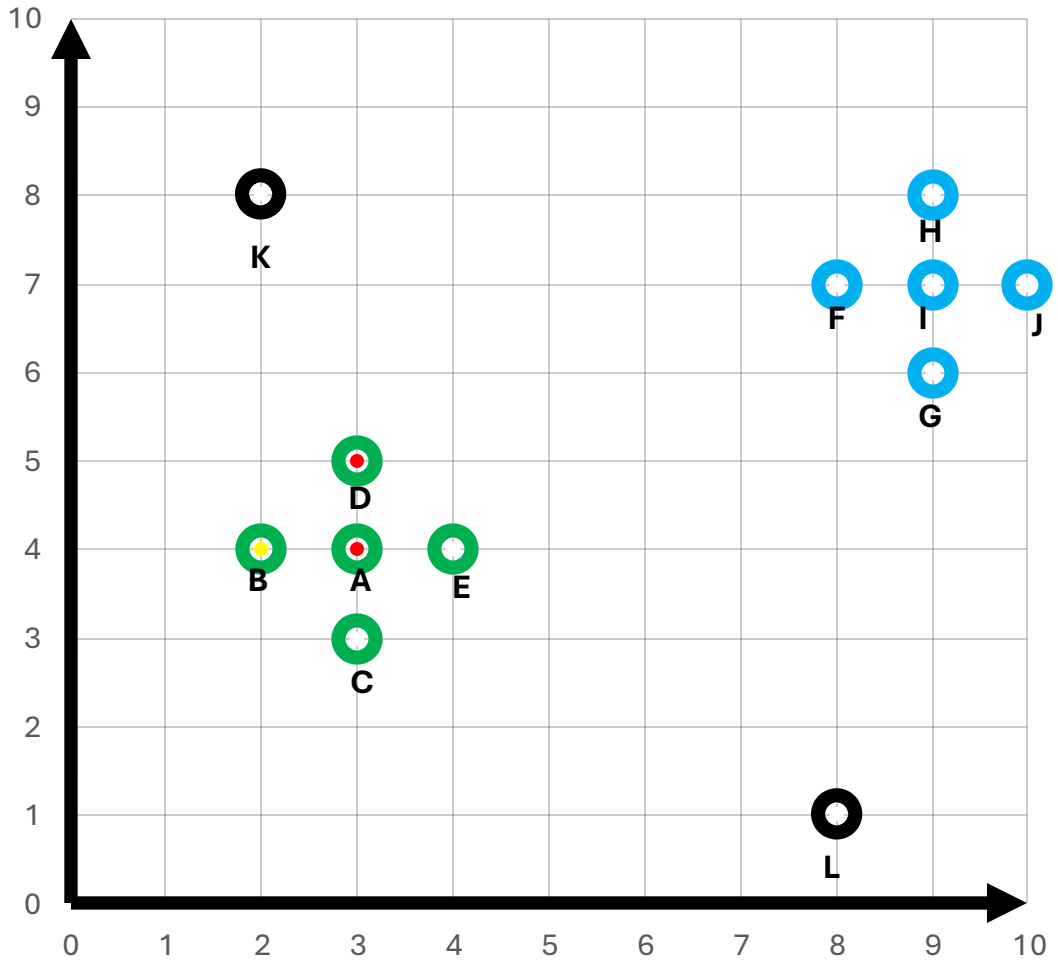


We then move to point **G**, **H** then **J**. Since they are all **non-core points**, we do not expand further.

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**

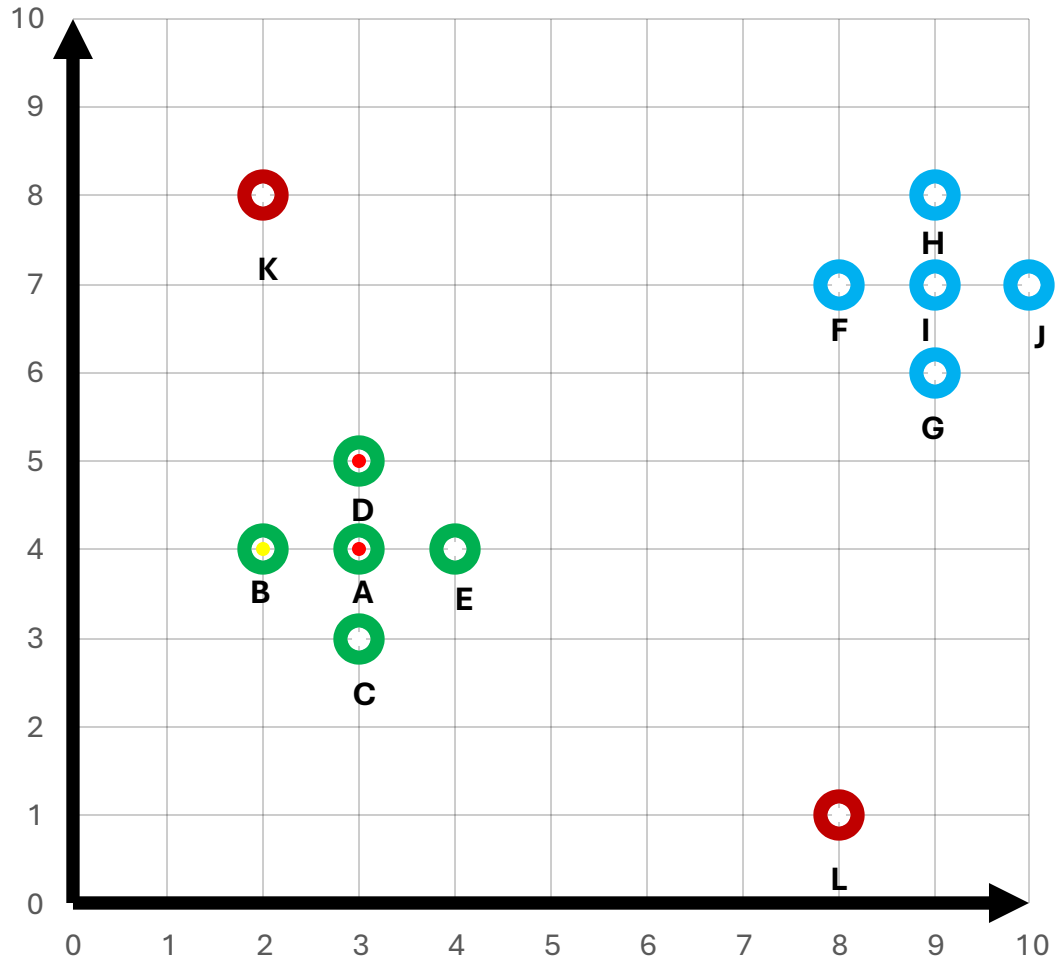


Clustering with DBSCAN is complete ☺



Note: If we were to have points **K** and **L** located at these coordinates

Customer	*Neighbours within eps*	*Number of neighbours*	Point Type	Cluster
A	B,C,D,E	4	**Core**	**GREEN**
B	A	1	**Non-core**	**GREEN**
C	A	1	**Non-core**	**GREEN**
D	A	1	**Non-core**	**GREEN**
E	I	1	**Non-core**	**GREEN**
F	I	1	**Non-core**	**BLUE**
G	I	1	**Non-core**	**BLUE**
H	I	1	**Non-core**	**BLUE**
I	F,G,H,J	4	**Core**	**BLUE**
J	I	1	**Non-core**	**BLUE**
K	None	0	**Noise**	None
L	None	0	**Noise**	None



Then we would label **K** and **L** as **noise points** which we also call **outliers**.