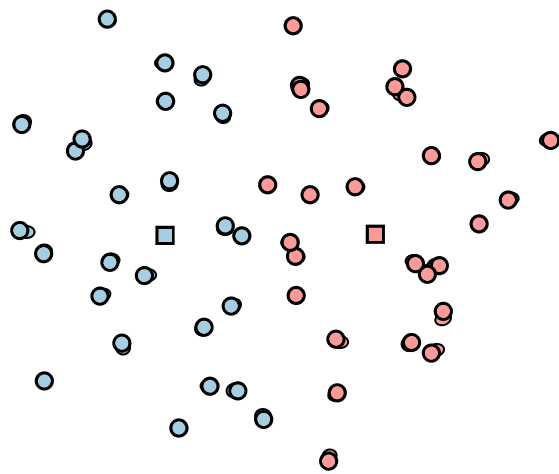
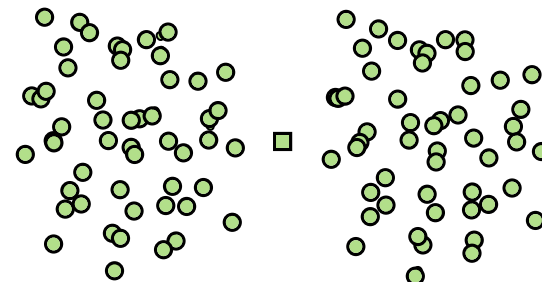
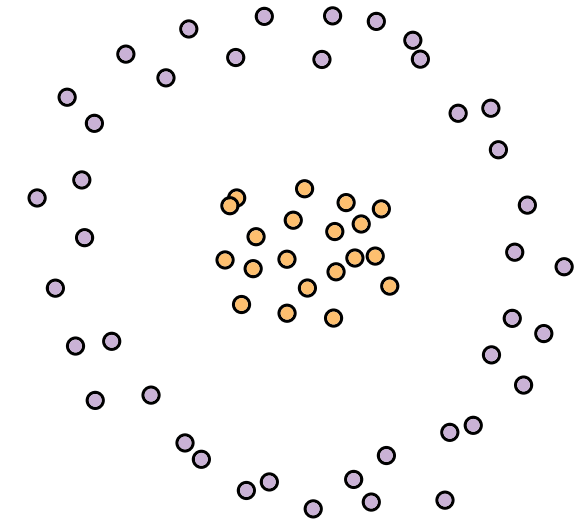


Algorithmen für geographische Informationssysteme

13. Vorlesung Clustering



Teil I: Problemstellung



Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

Fundamentales Problem im Data Mining, aber nicht eindeutig definiert.

Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

Fundamentales Problem im Data Mining, aber nicht eindeutig definiert.

- Was willst du clustern?

Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

Fundamentales Problem im Data Mining, aber nicht eindeutig definiert.

- Was willst du clustern?
- Was willst du mit den Daten tun?

Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

Fundamentales Problem im Data Mining, aber nicht eindeutig definiert.

- Was willst du clustern?
- Was willst du mit den Daten tun?
- Distanzen:
 - Euklidisch?
 - metrisch?

Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

Fundamentales Problem im Data Mining, aber nicht eindeutig definiert.

- Was willst du clustern?
- Was willst du mit den Daten tun?
- Distanzen:
 - Euklidisch?
 - metrisch?
- Wie viele Cluster?

Clustering



Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.

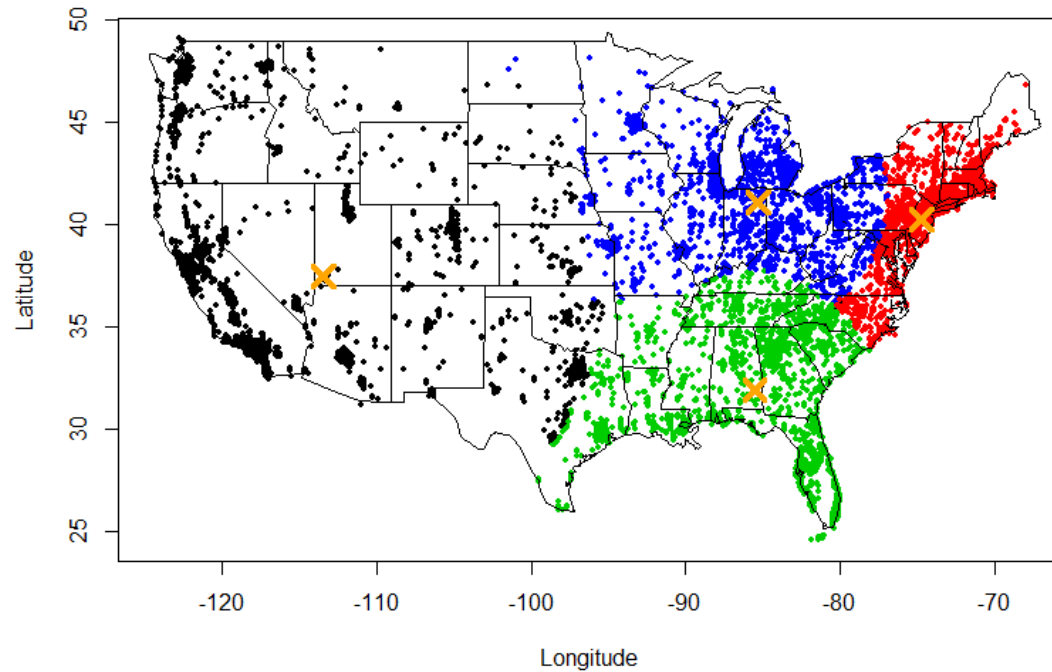
Fundamentales Problem im Data Mining, aber nicht eindeutig definiert.

- Was willst du clustern?
- Was willst du mit den Daten tun?
- Distanzen:
 - Euklidisch?
 - metrisch?
- Wie viele Cluster?
- Wie kann ein Cluster aussehen?

Clustering



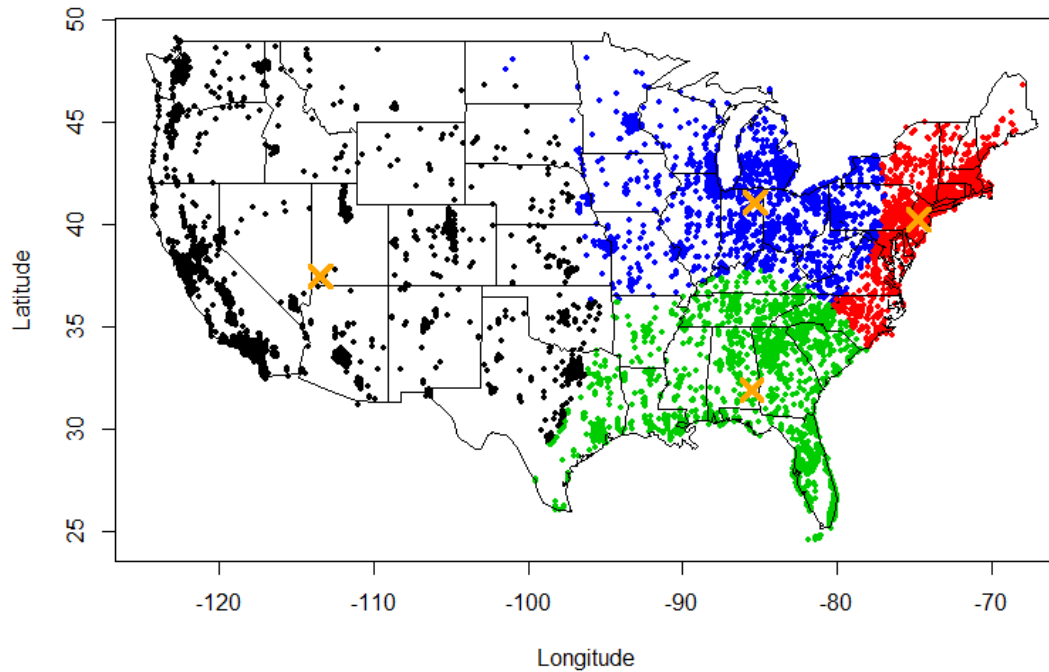
Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.



Clustering



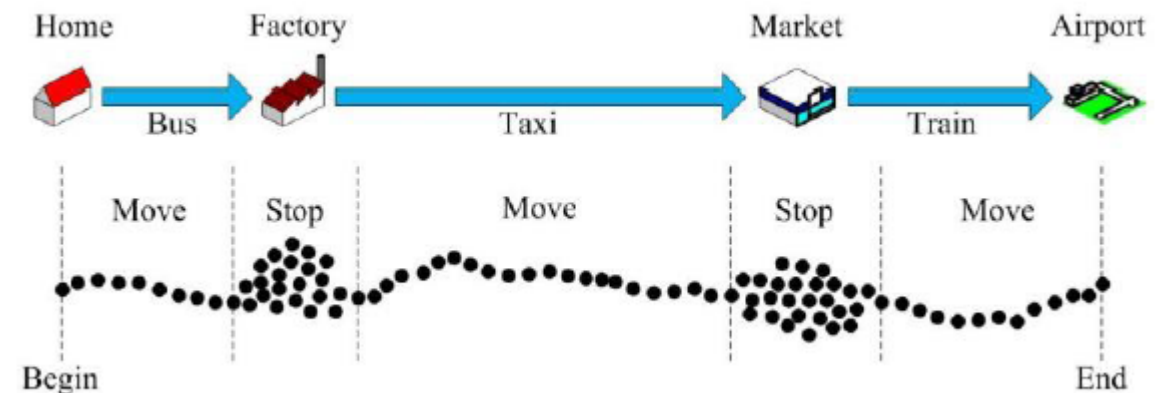
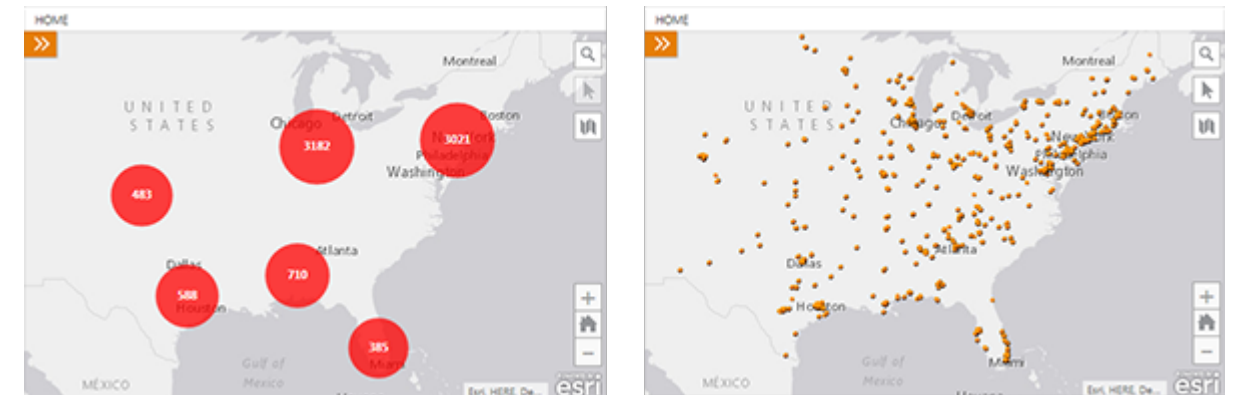
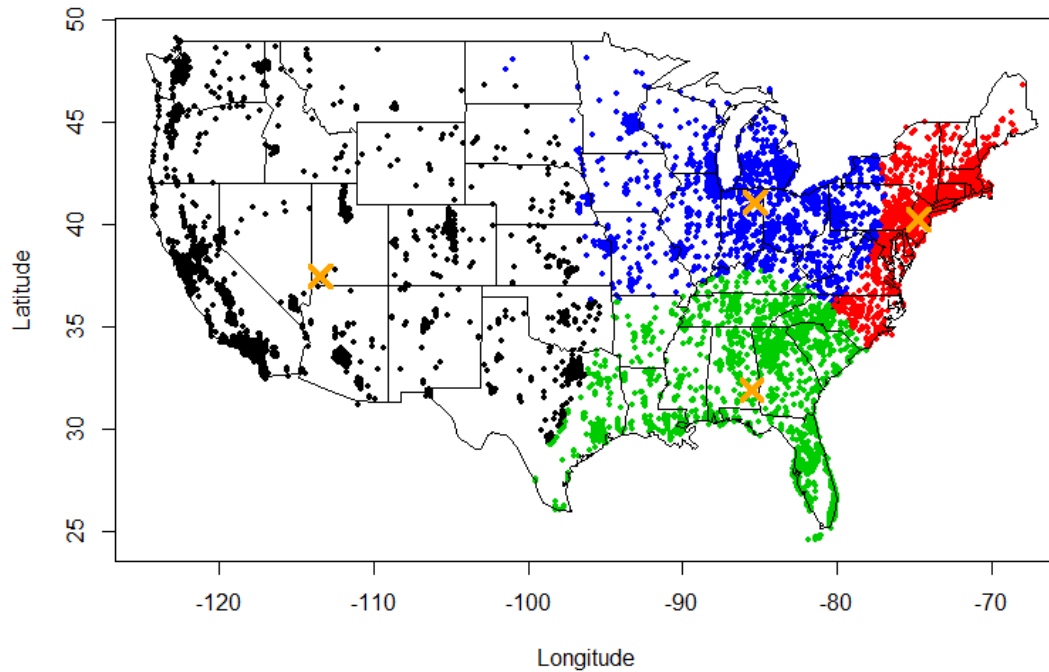
Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.



Clustering



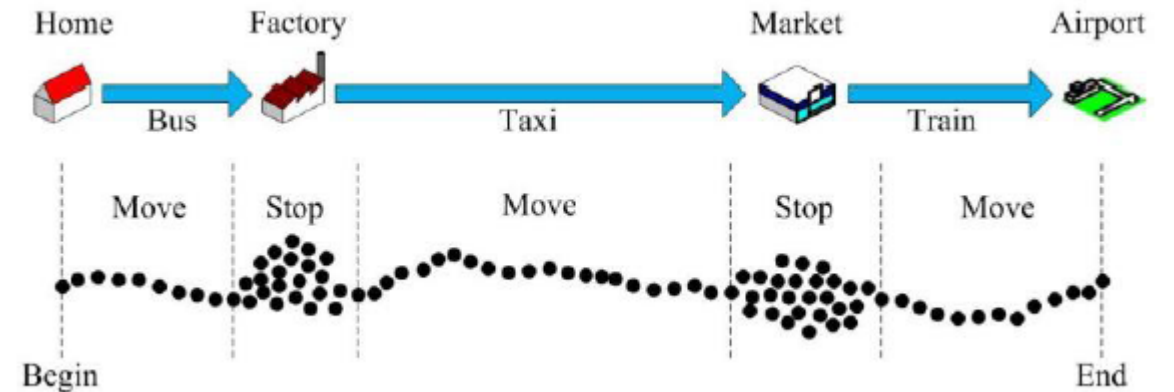
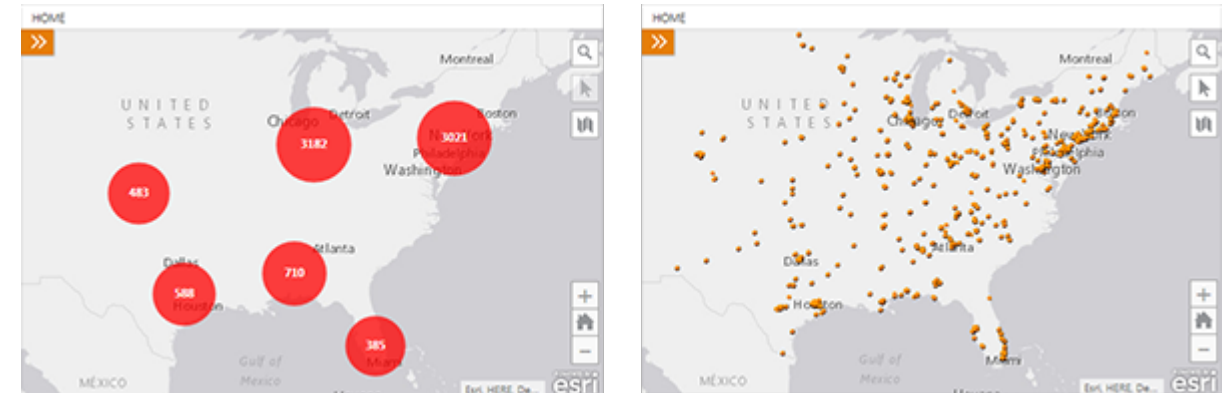
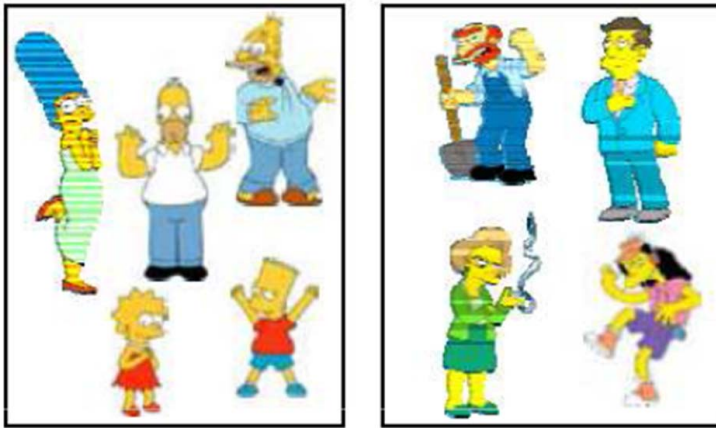
Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.



Clustering

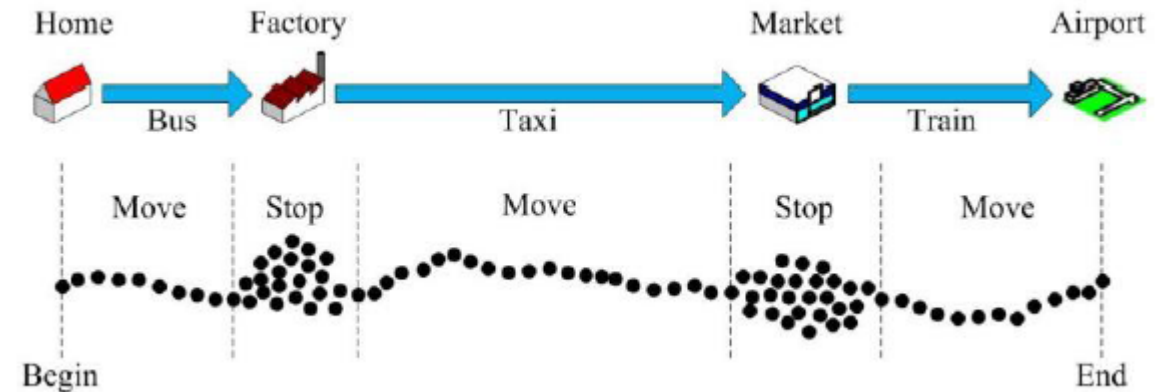
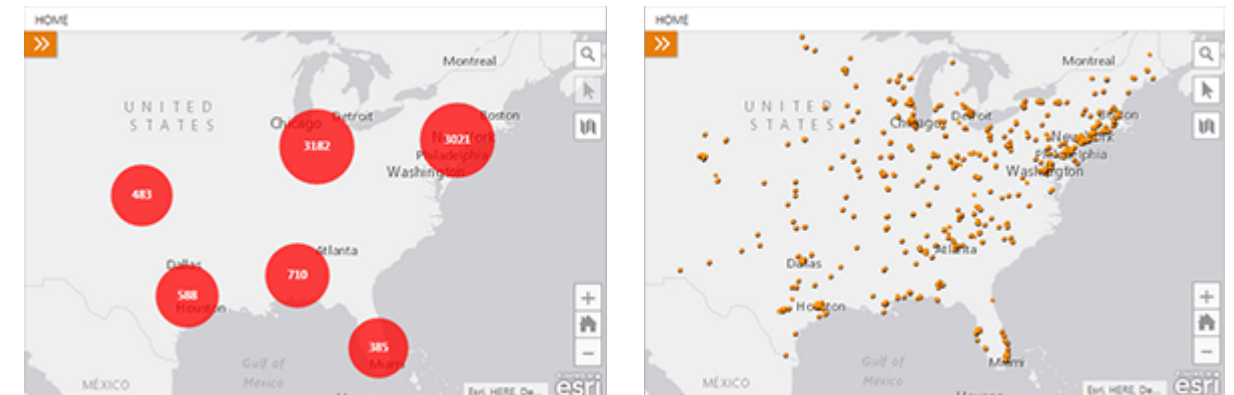
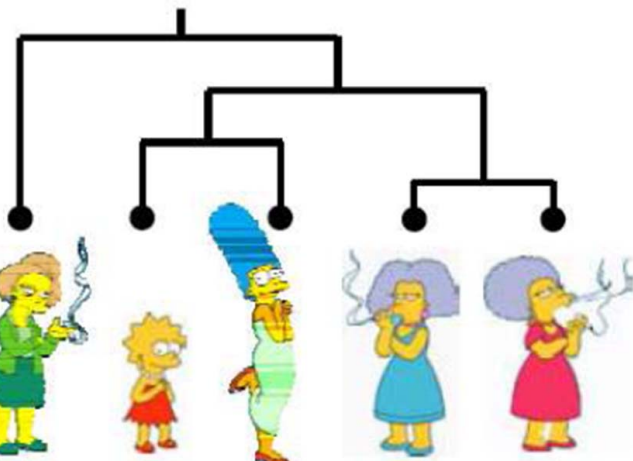
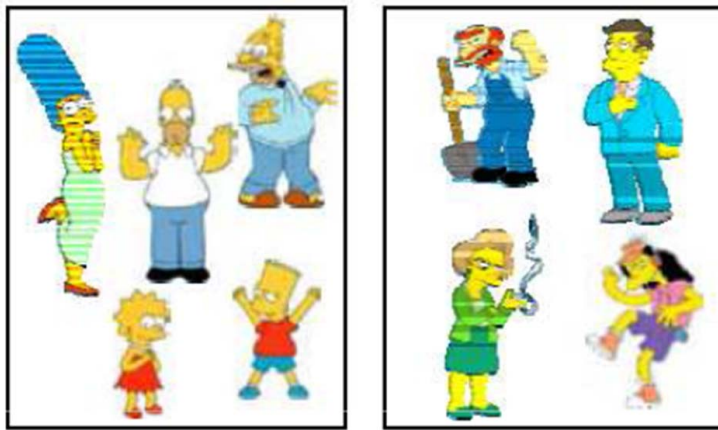


Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.



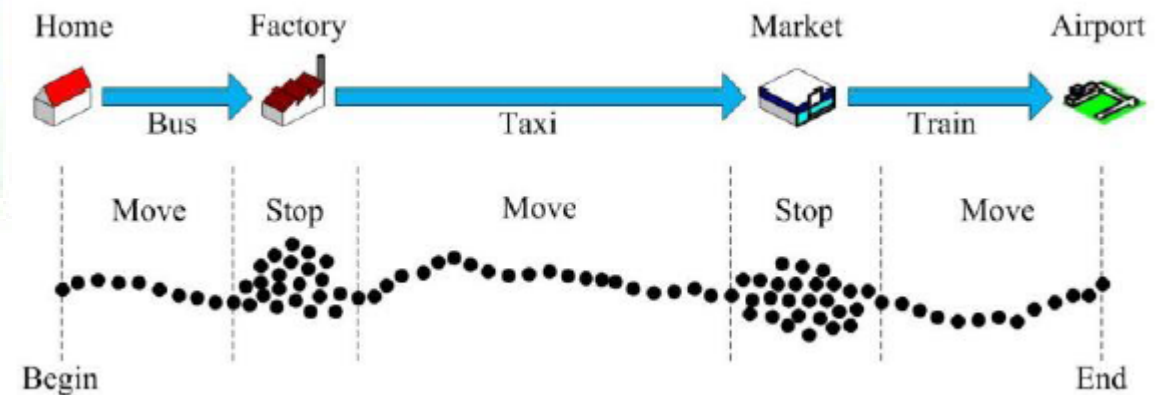
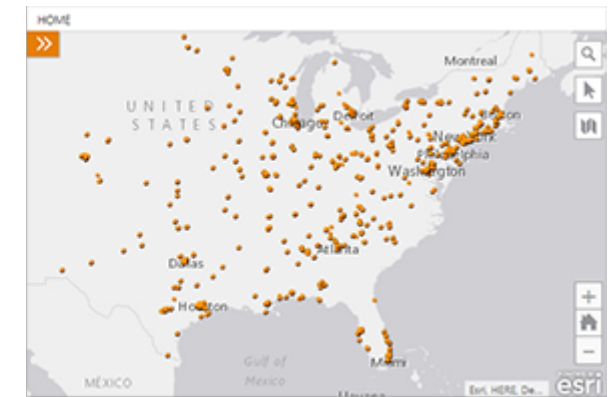
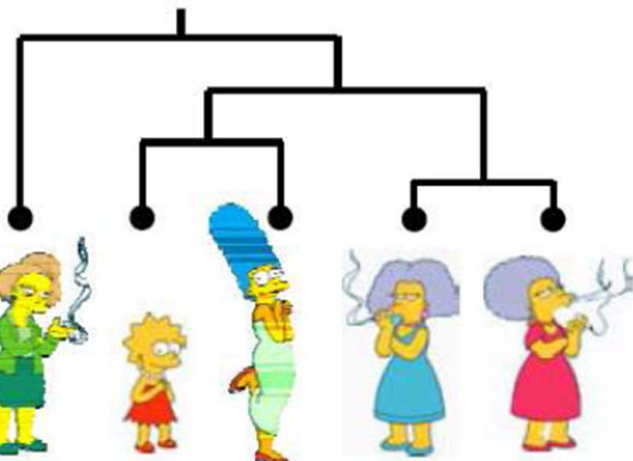
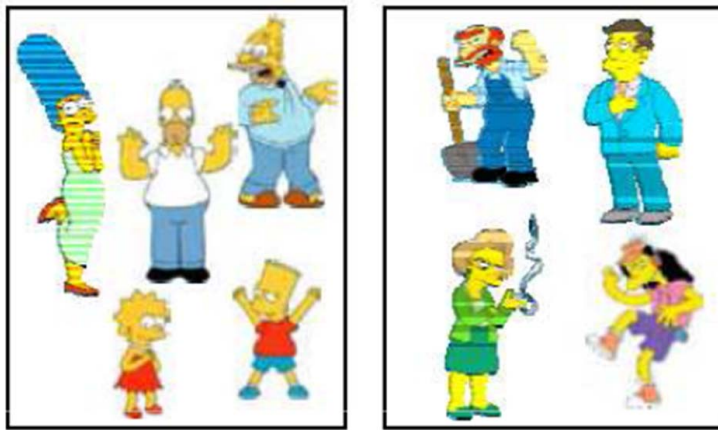
Clustering

Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.



Clustering

Clustering ist klassischerweise das Problem, eine Partition eines Datensatzes zu finden, sodass Elemente in derselben Zelle („Cluster“) gemäß einem gegebenen Distanzkriterium nahe beieinander liegen, während Elemente in verschiedenen Mengen weit voneinander entfernt sind.



[Steinhaus 1956]

MATHÉMATIQUE

Sur la division des corps matériels en parties

par

H. STEINHAUS

Présenté le 19 Octobre 1956

Un corps Q est, par définition, une répartition de matière dans l'espace, donnée par une fonction $f(P)$; on appelle cette fonction **la densité** du corps en question; elle est définie pour tous les points P de l'espace; elle est non-négative et mesurable. On suppose que l'ensemble caractérisant le corps $E = \{P \mid f(P) > 0\}$ est borné et de mesure positive; on

k-MEANS

[MacQueen 1967]

SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS

J. MACQUEEN

UNIVERSITY OF CALIFORNIA, LOS ANGELES

1. Introduction

The main purpose of this paper is to describe a process for partitioning an N -dimensional population into k sets on the basis of a sample. The process, which is called '*k*-means,' appears to give partitions which are reasonably efficient in the sense of within-class variance. That is, if p is the probability mass function for the population, $S = \{S_1, S_2, \dots, S_k\}$ is a partition of E_N , and u_i ,

Clustering



DBSCAN

[Ester, Kriegel, Sandra & Xu 1996]

A **D**ensity-**B**ased Algorithm for Discovering **C**lusters in Large Spatial Databases with **N**oise?

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 München, Germany
{ester | kriegel | sander | xwxu}@informatik.uni-muenchen.de

Clustering



DBSCAN

[Ester, Kriegel, Sandra & Xu 1996]

A **D**ensity-**B**ased Algorithm for Discovering **C**lusters in Large Spatial Databases with **N**oise?

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 München, Germany
{ester | kriegel | sander | xwxu}@informatik.uni-muenchen.de

> 8000 Zitate

KDD “test of time award” 2014

Open Source Implementierungen verfügbar in vielen Sprachen

DBSCAN

[Ester, Kriegel, Sandra & Xu 1996]

A **D**ensity-**B**ased Algorithm for Discovering **C**lusters in Large Spatial Databases with **N**oise?

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

Institute for Computer Science, University of Munich

Oettingenstr. 67, D-80538 München, Germany

{ester | kriegel | sander | xwxu}@informatik.uni-muenchen.de

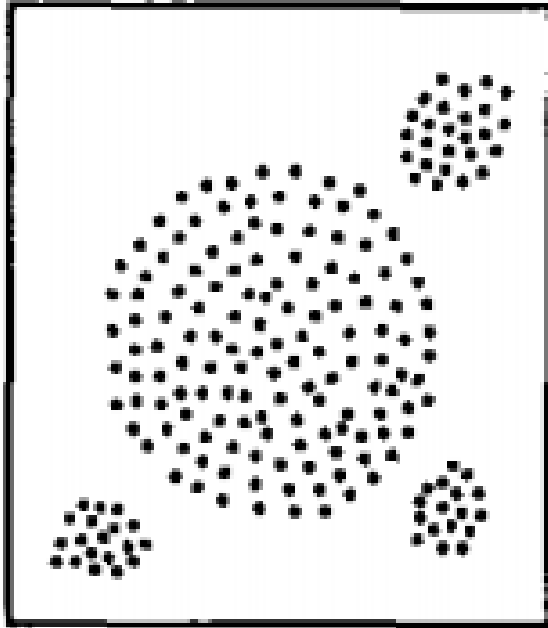
ing an appropriate value for it. It discovers clusters of arbitrary shape. Finally, DBSCAN is efficient even for large spatial databases. The rest of the paper is organized as follows.

> 8000 Zitate

KDD “test of time award” 2014

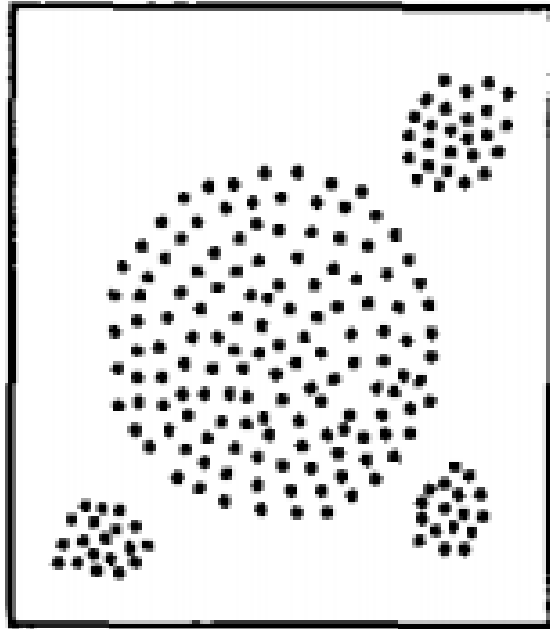
Open Source Implementierungen verfügbar in vielen Sprachen

Clustering

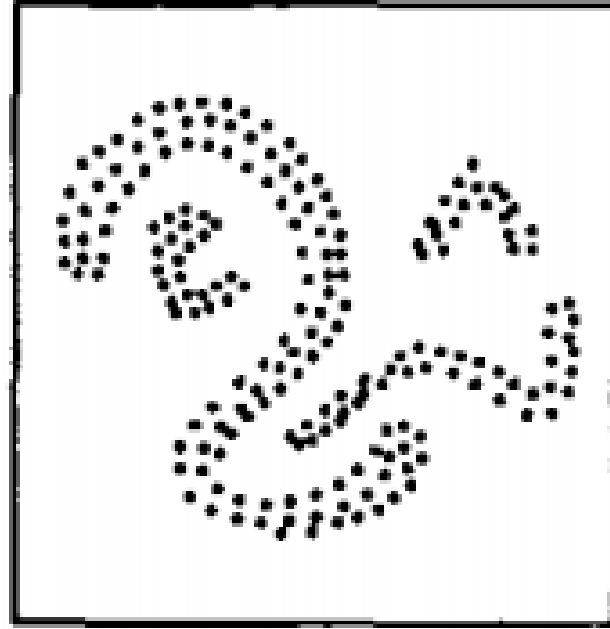


database 1

Clustering

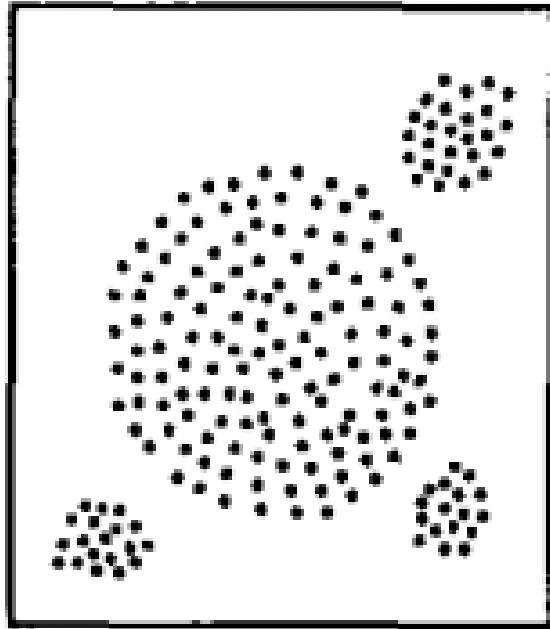


database 1

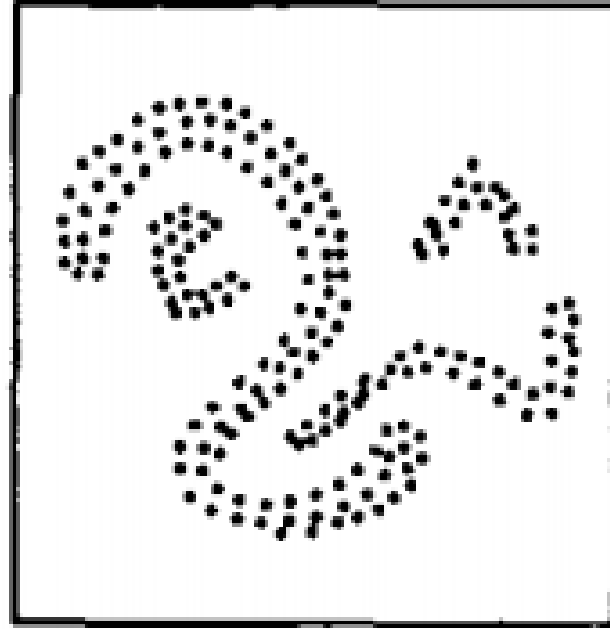


database 2

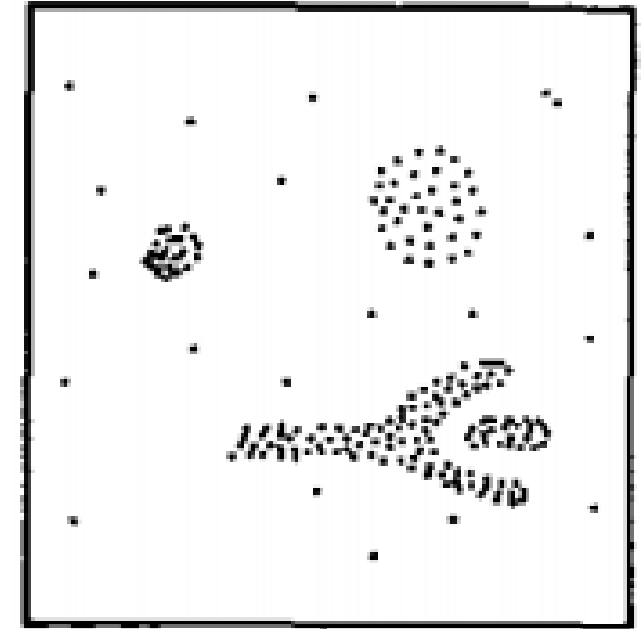
Clustering



database 1



database 2



database 3

Zielfunktionen



1. „Minimale Anforderungen an Domänenwissen zur Bestimmung der Eingabeparameter, da geeignete Werte oft nicht im Voraus bekannt sind, wenn mit großen Datenbanken gearbeitet wird.“

Zielfunktionen



1. „Minimale Anforderungen an Domänenwissen zur Bestimmung der Eingabeparameter, da geeignete Werte oft nicht im Voraus bekannt sind, wenn mit großen Datenbanken gearbeitet wird.“
2. „Entdeckung von Clustern mit beliebiger Form, da die Form von Clustern in räumlichen Datenbanken kugelförmig, langgezogen, linear, verlängert etc. sein kann.“

Zielfunktionen

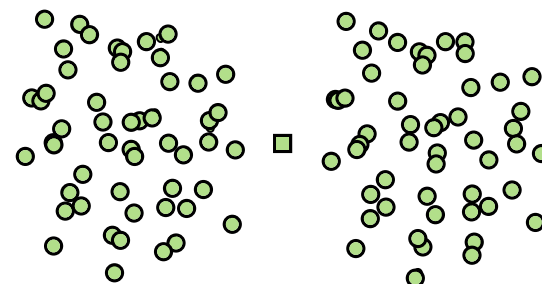
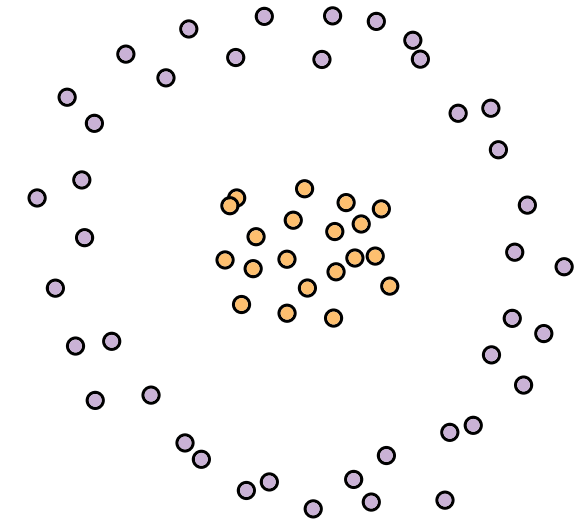
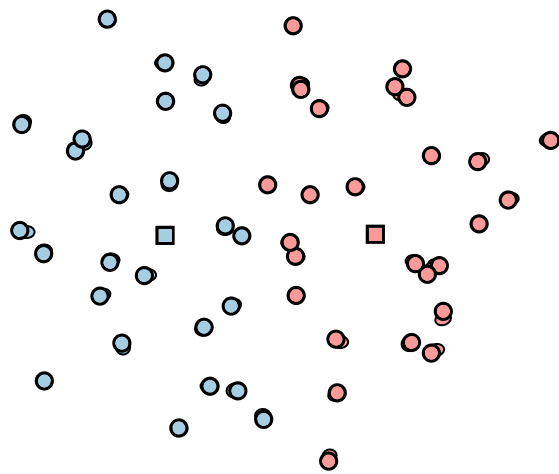


1. „Minimale Anforderungen an Domänenwissen zur Bestimmung der Eingabeparameter, da geeignete Werte oft nicht im Voraus bekannt sind, wenn mit großen Datenbanken gearbeitet wird.“
2. „Entdeckung von Clustern mit beliebiger Form, da die Form von Clustern in räumlichen Datenbanken kugelförmig, langgezogen, linear, verlängert etc. sein kann.“
3. „Gute Effizienz bei großen Datenbanken, d.h. bei Datenbanken mit deutlich mehr als nur ein paar tausend Objekten.“

Algorithmen für geographische Informationssysteme

13. Vorlesung Clustering

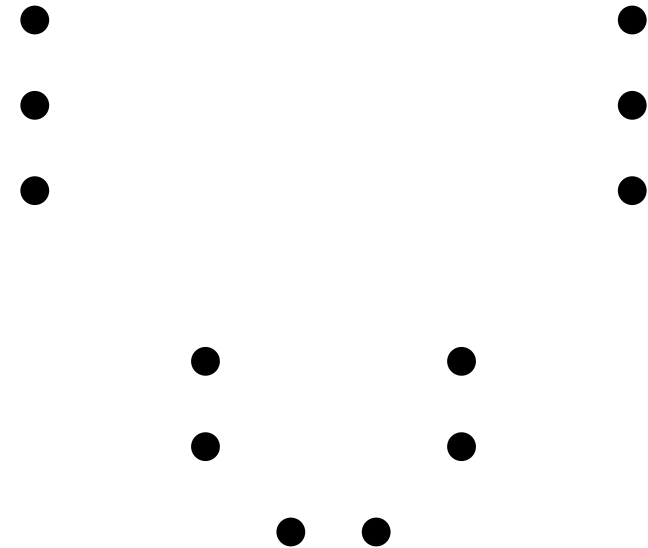
Teil II: k -MEANS



k -MEANS



Iterative Heuristik.

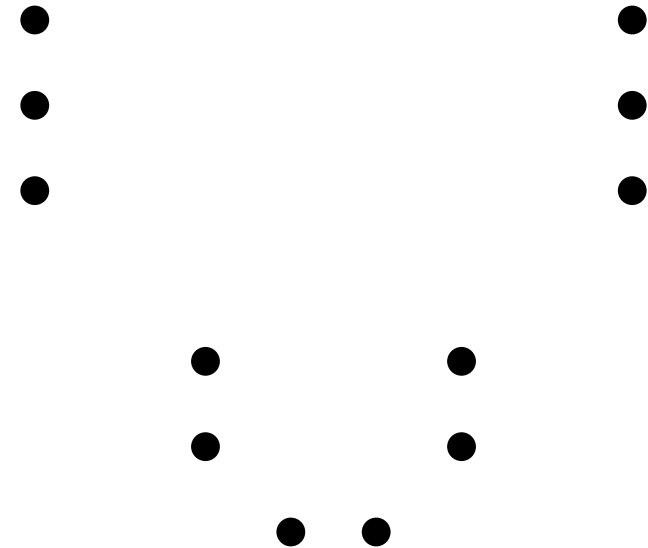


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren

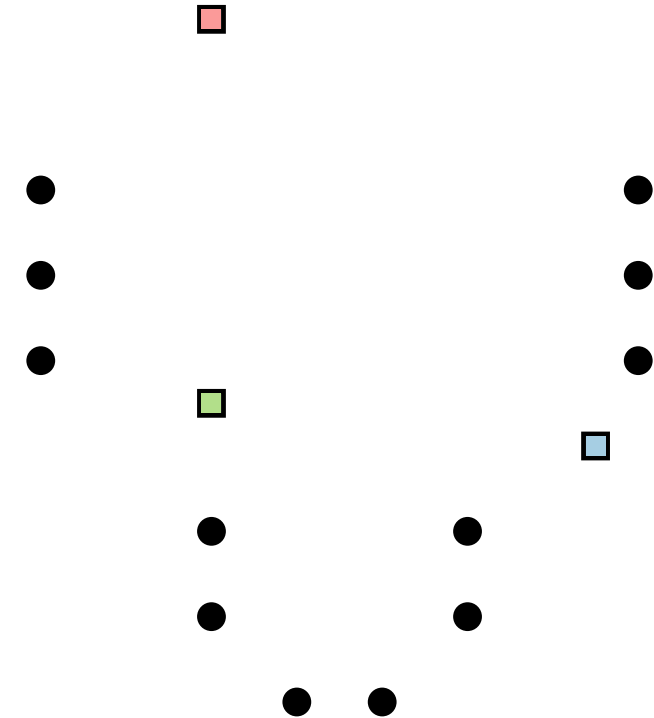


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren

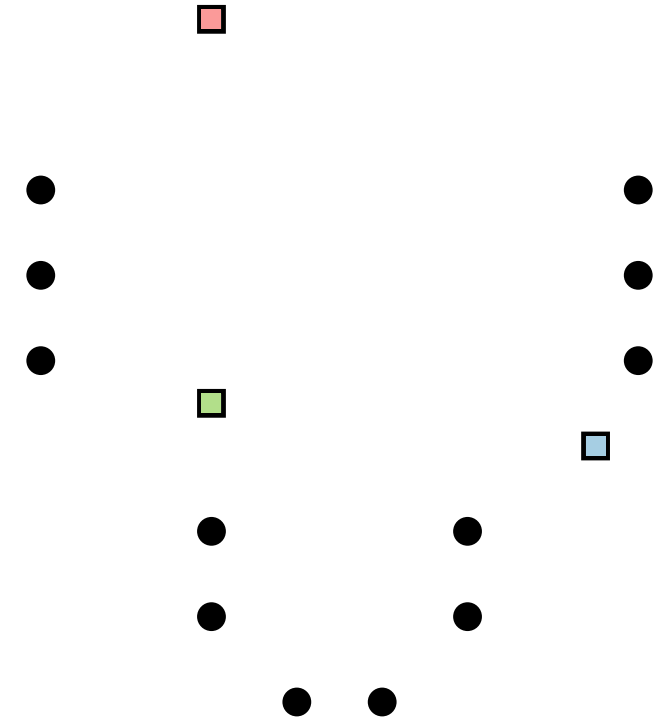


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu

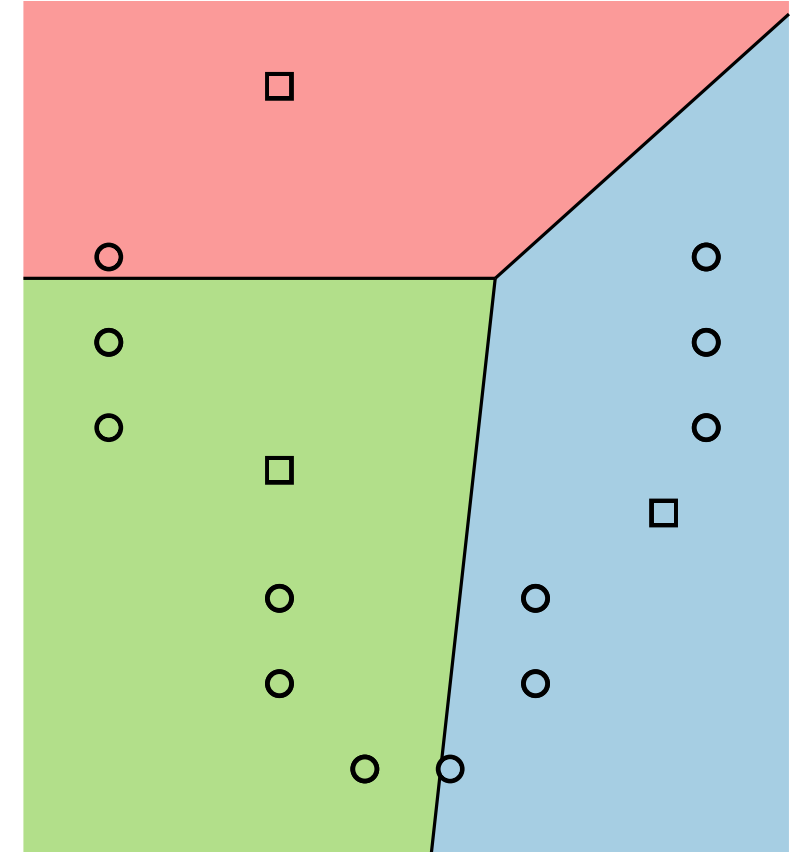


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu

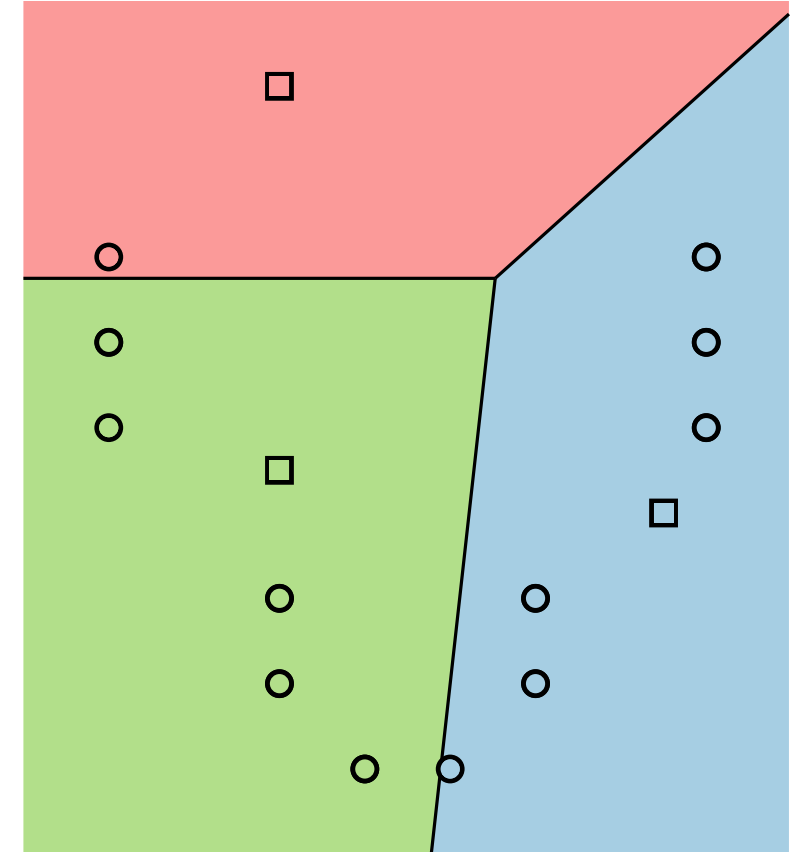


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

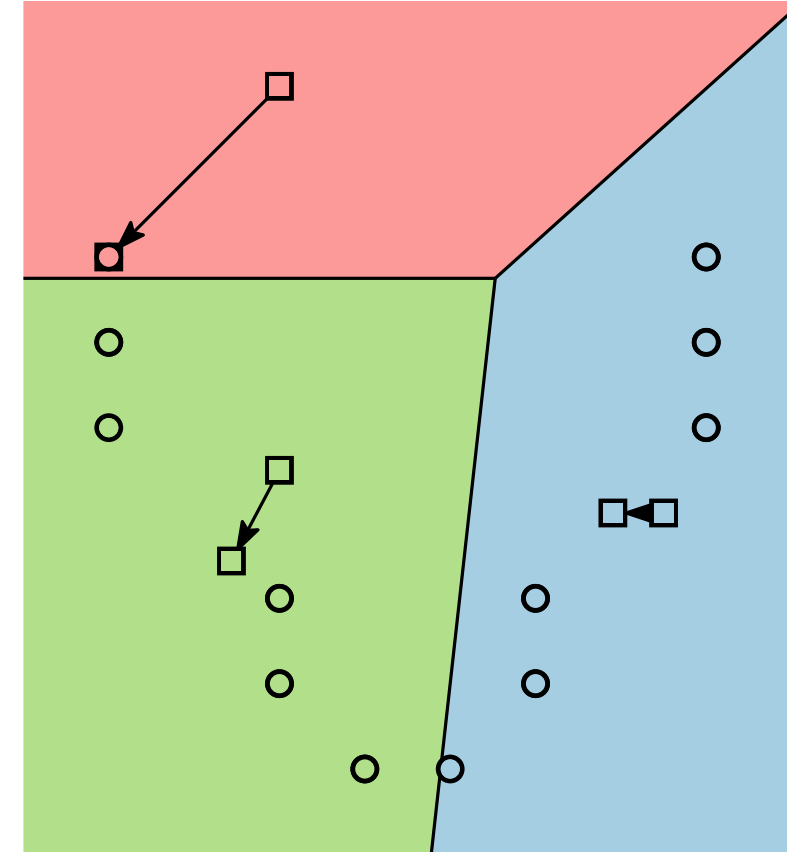


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

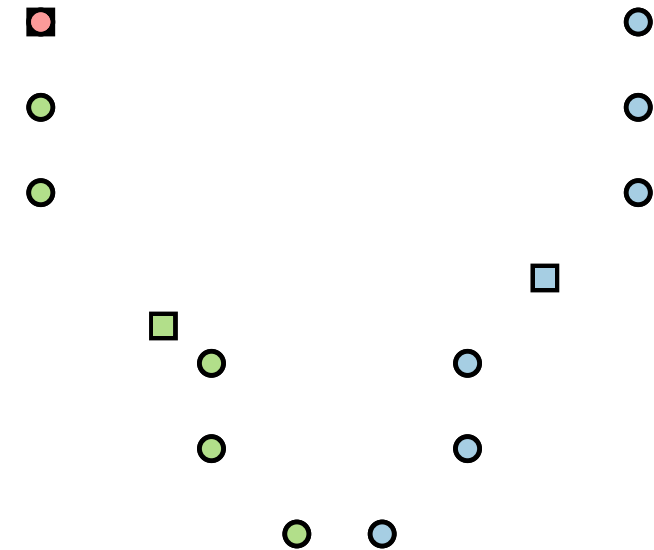


k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters



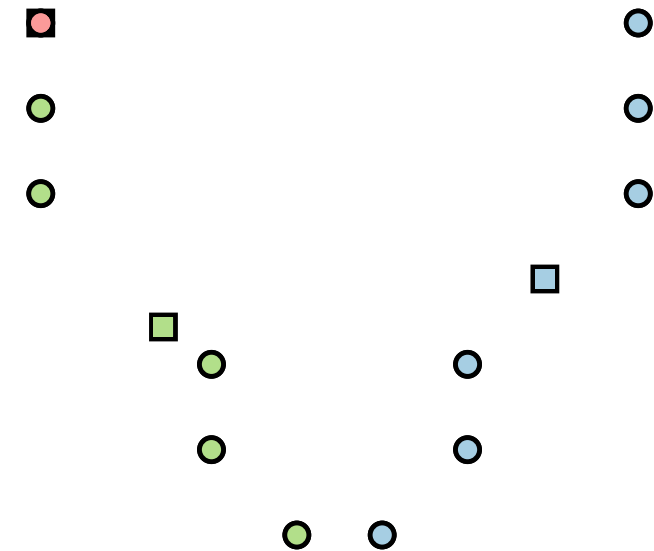
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



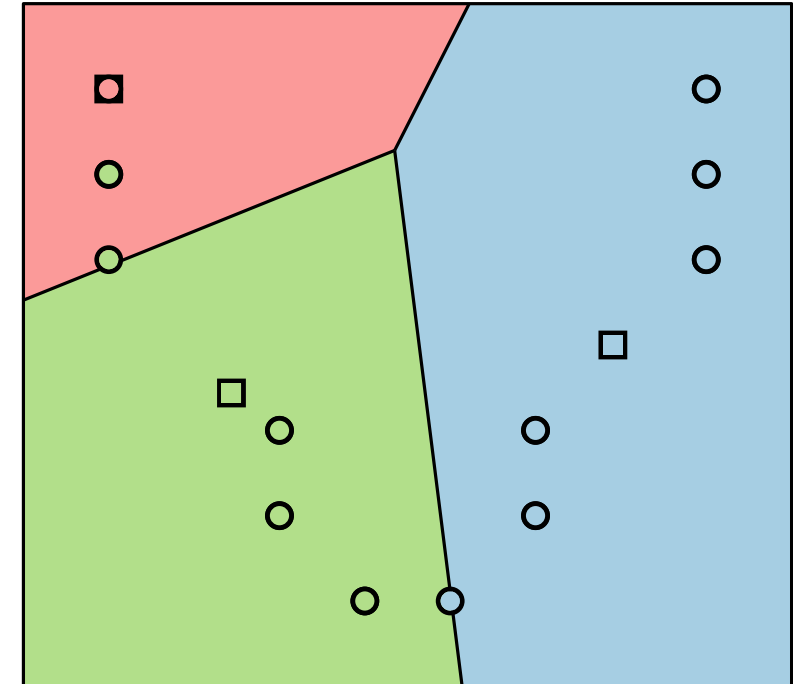
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



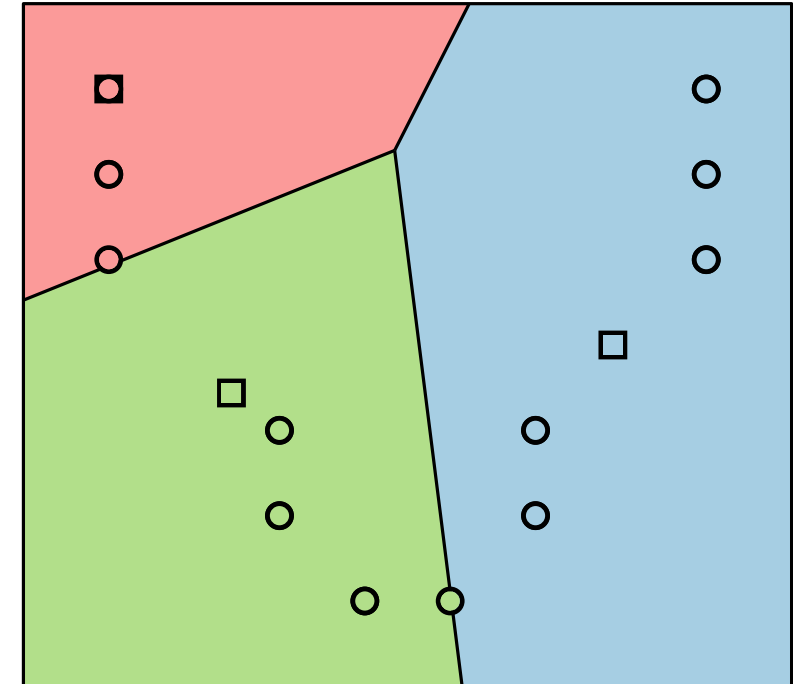
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



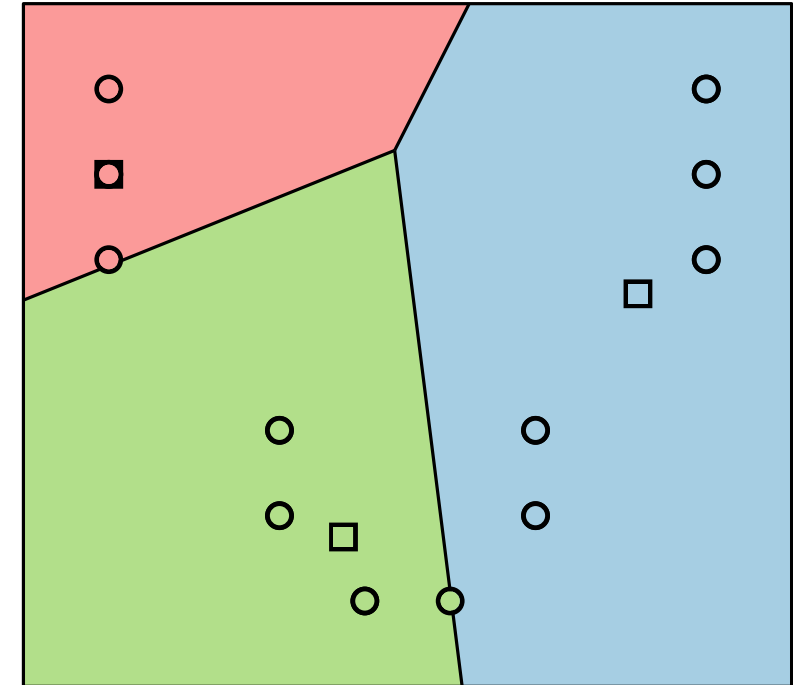
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



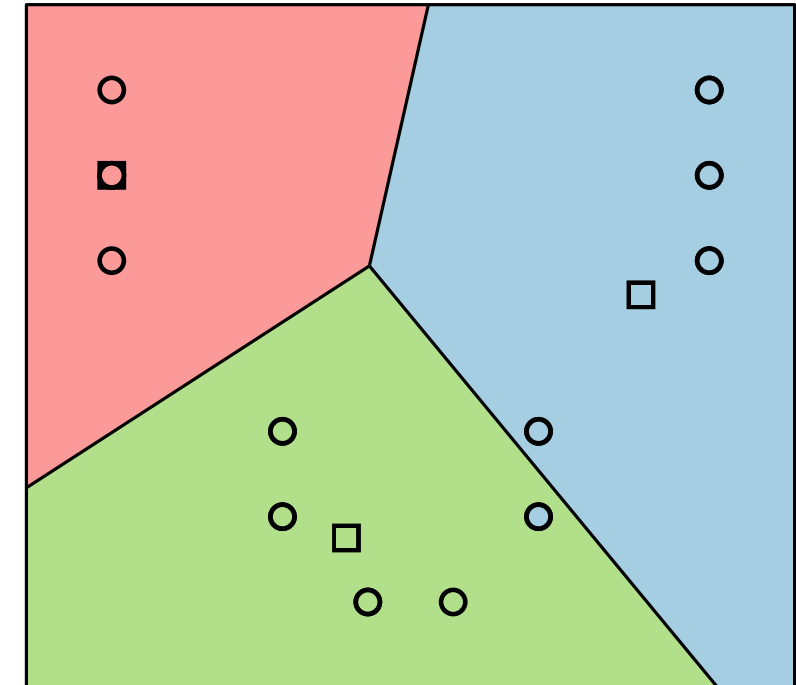
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



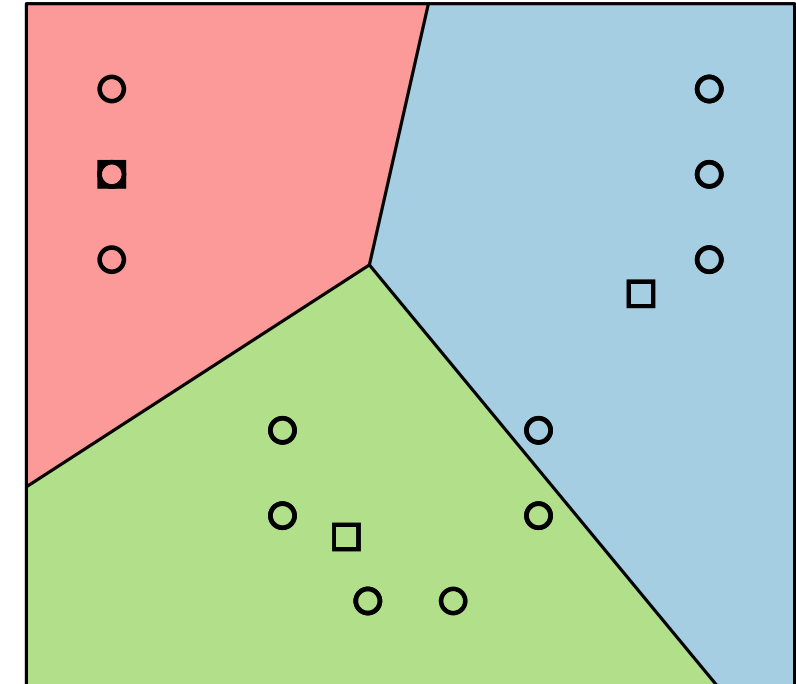
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



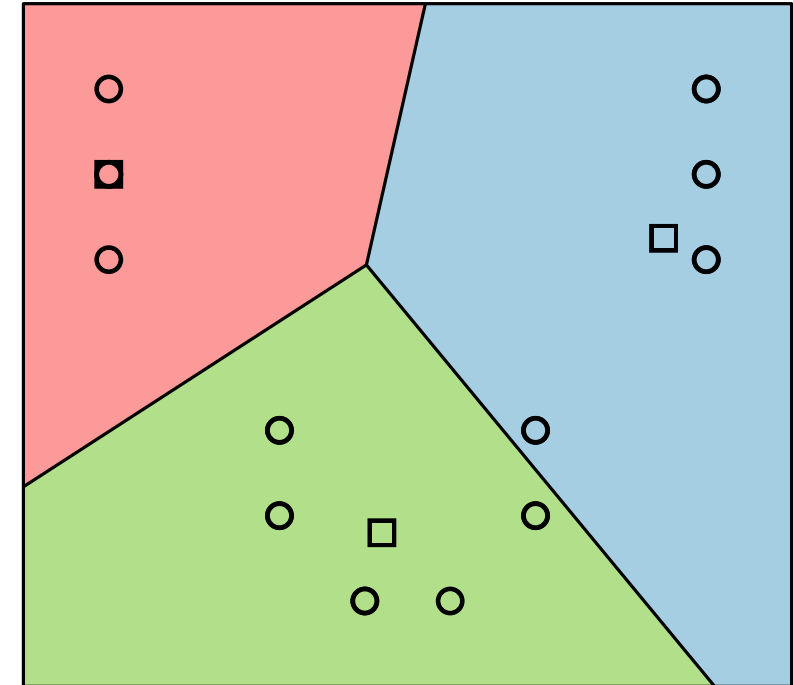
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



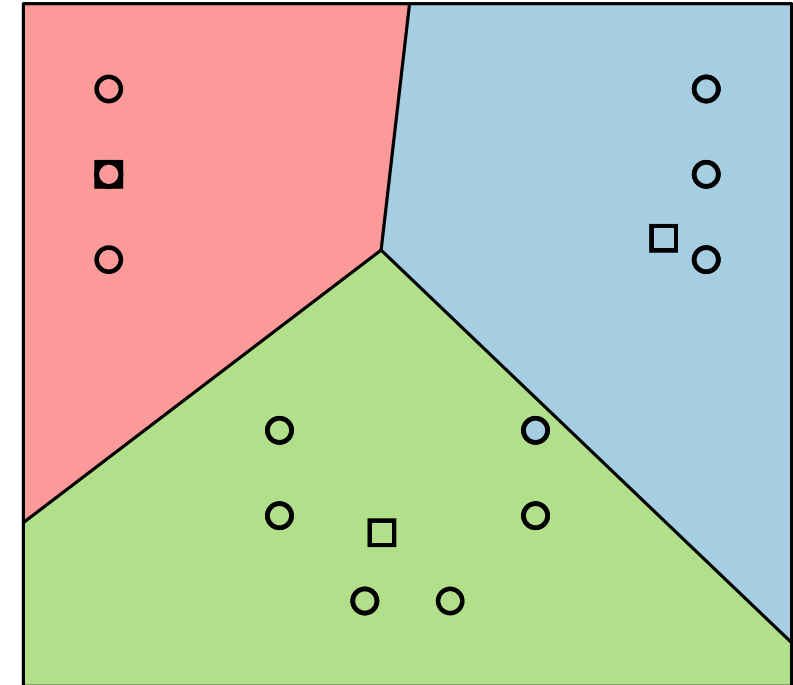
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



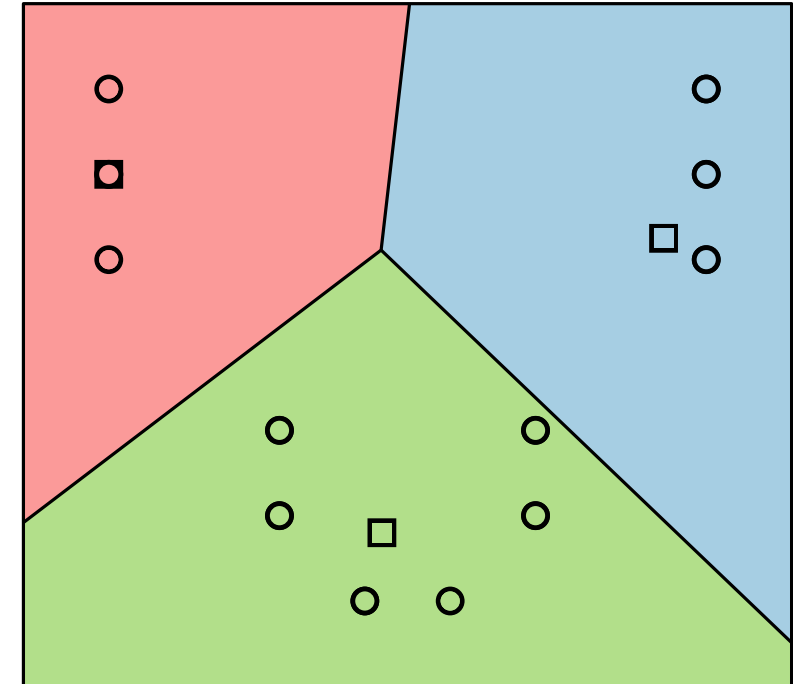
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



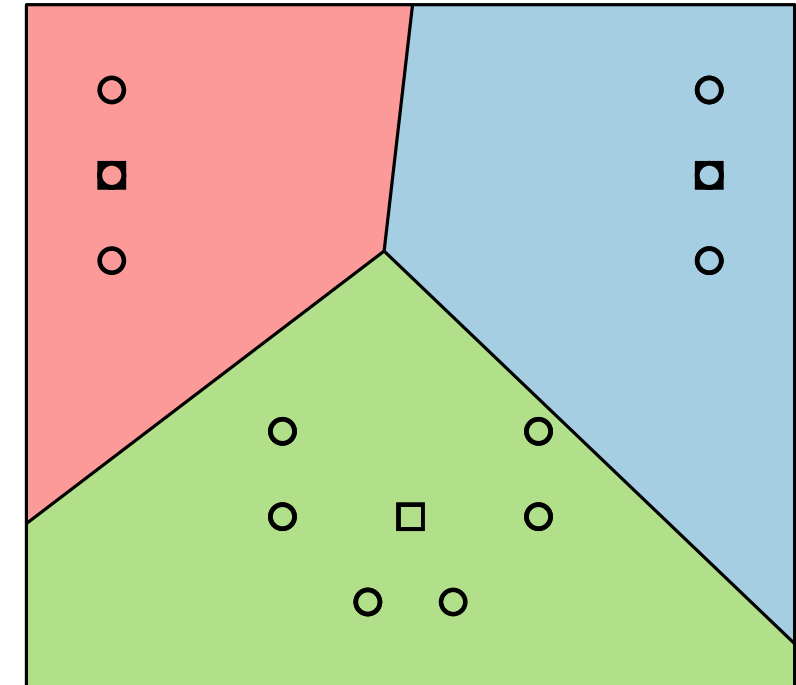
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



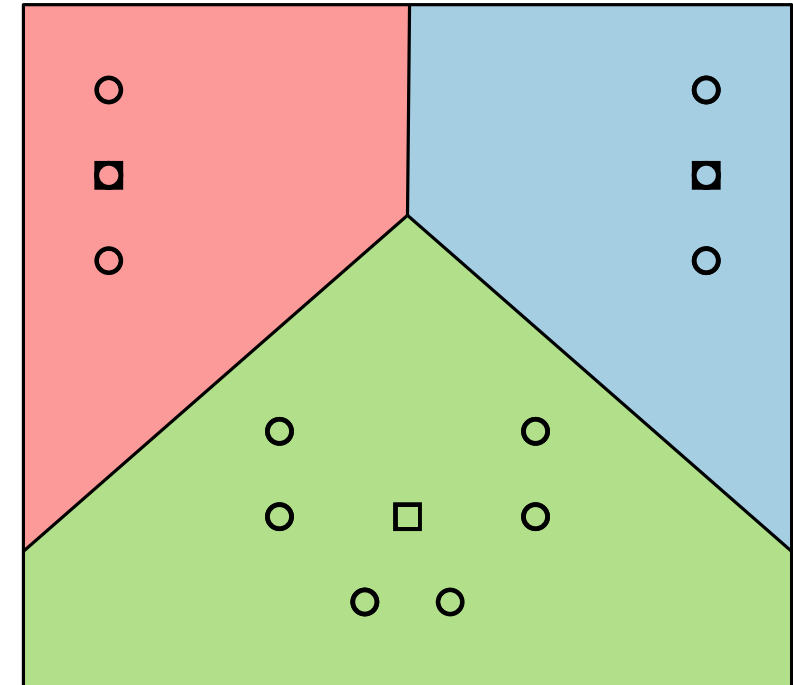
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



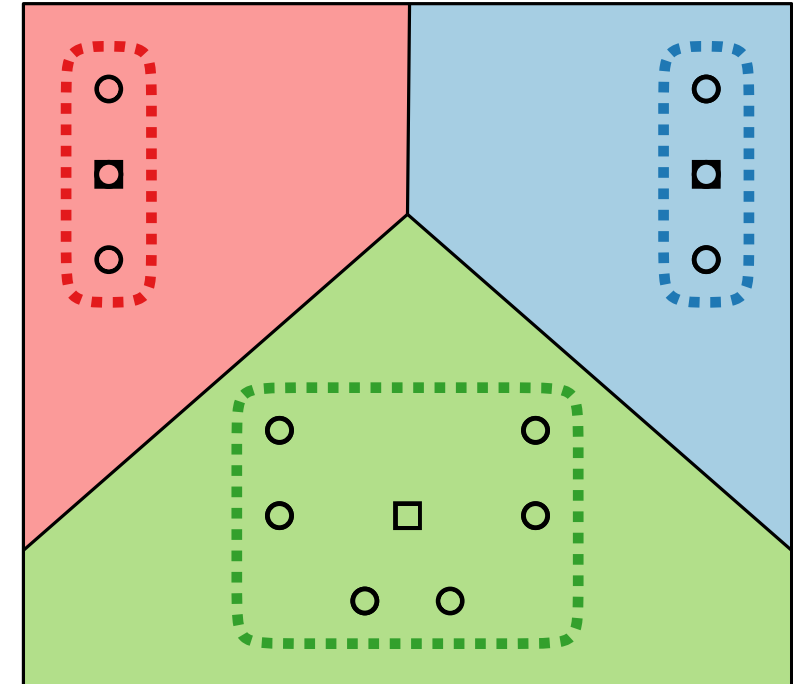
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



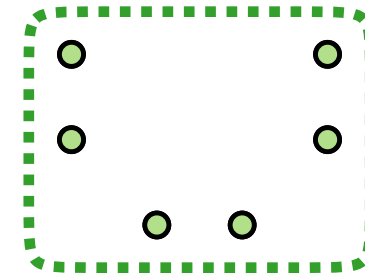
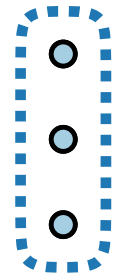
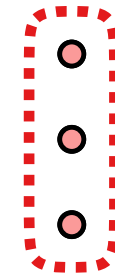
k -MEANS



Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert



k -MEANS

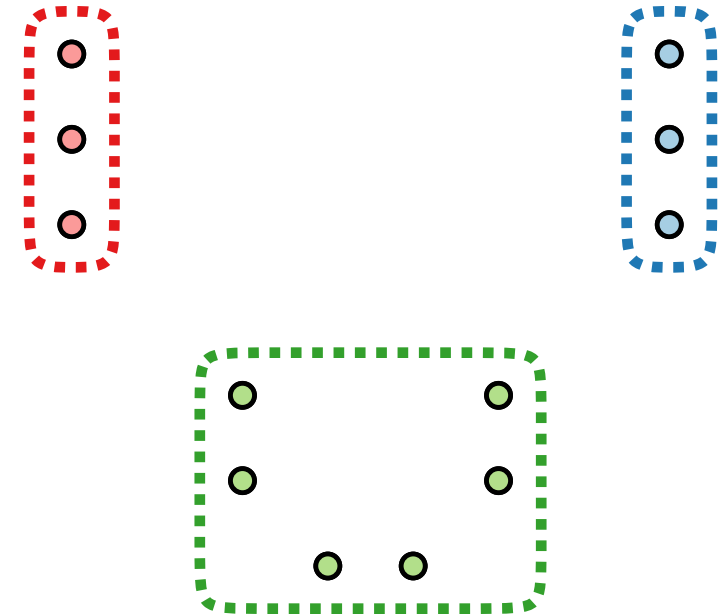


Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert

Laufzeit.



k -MEANS



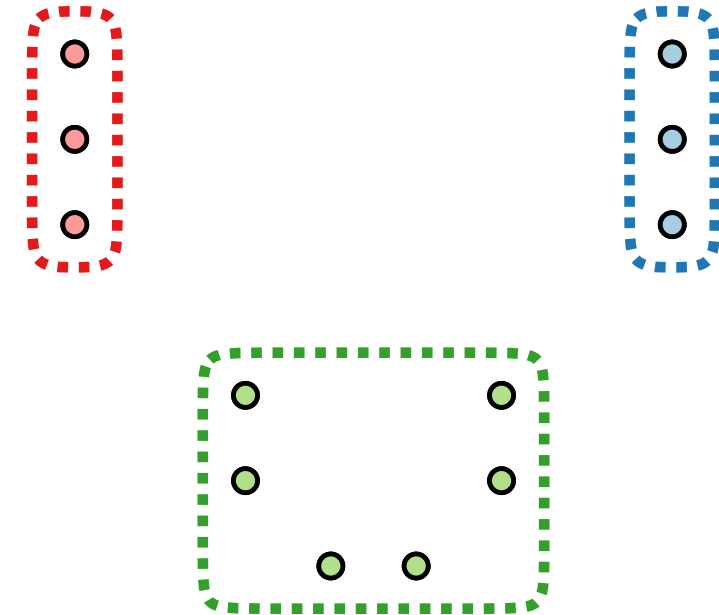
Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert

Laufzeit.

1. $\mathcal{O}(k)$



k -MEANS



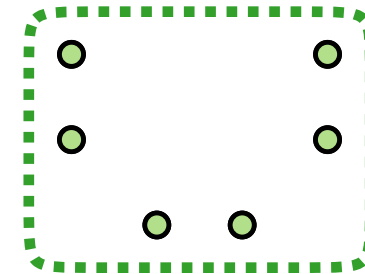
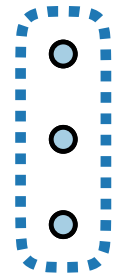
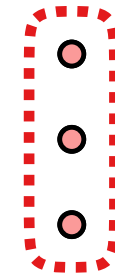
Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert

Laufzeit.

1. $\mathcal{O}(k)$
2. $\mathcal{O}(kn)$ pro Iteration



k -MEANS



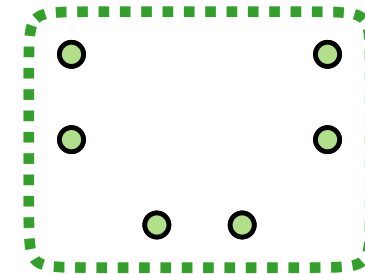
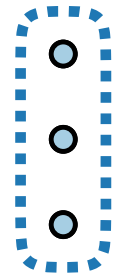
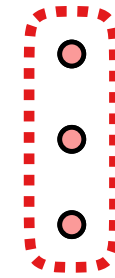
Iterative Heuristik.

1. Wähle k zufällige Punkte als Clusterzentren
2. Weiße Punkte nächsten Clusterzentren zu
3. Verschiebe Clusterzentren auf Schwerpunkt des Clusters

Wiederhole 3. und 4. bis sich nichts mehr verändert

Laufzeit.

1. $\mathcal{O}(k)$
2. $\mathcal{O}(kn)$ pro Iteration
3. $\mathcal{O}(n)$ pro Iteration



k -MEANS: Beispiele



Segmentierung

Original



k -MEANS: Beispiele

Segmentierung

$k = 2$



Original



k -MEANS: Beispiele

Segmentierung

$k = 2$



$k = 3$



Original



k -MEANS: Beispiele

Segmentierung

$k = 2$



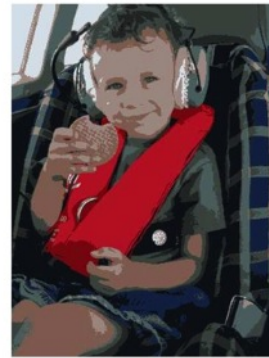
$k = 3$



$k = 10$



Original



k -MEANS: Beispiele



Segmentierung

$k = 2$



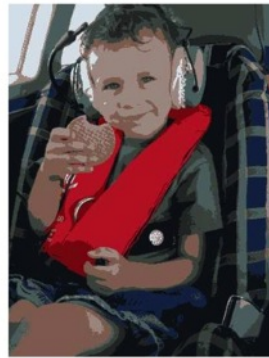
$k = 3$



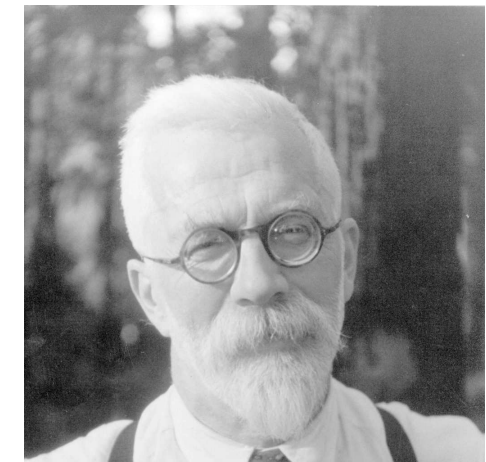
$k = 10$



Original



Vektorquantisierung



8 Bits / Pixel

k -MEANS: Beispiele



Segmentierung

$k = 2$



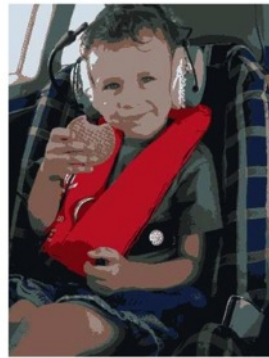
$k = 3$



$k = 10$

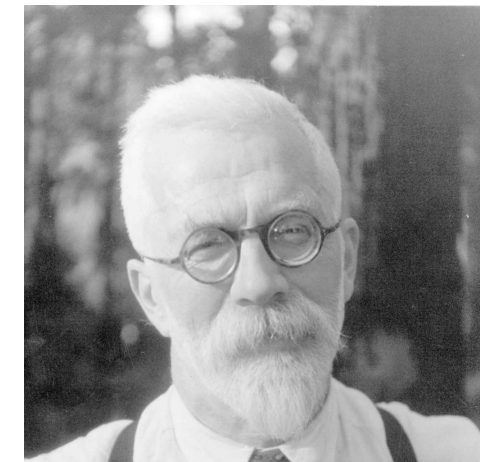
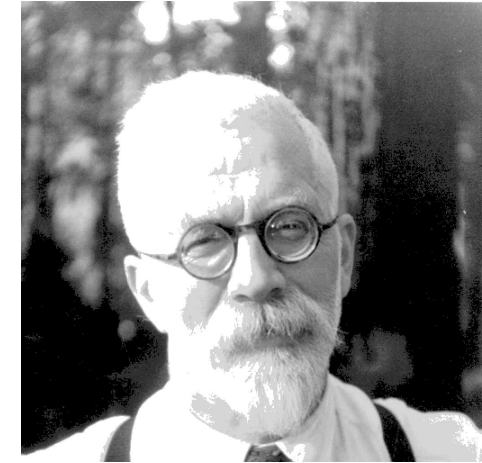


Original



Vektorquantisierung

1,9 Bits / Pixel



8 Bits / Pixel

k -MEANS: Beispiele



Segmentierung

$k = 2$



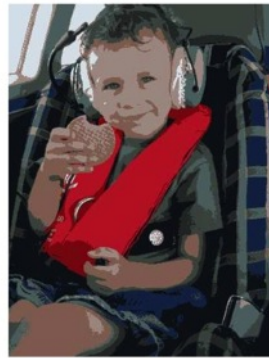
$k = 3$



$k = 10$



Original

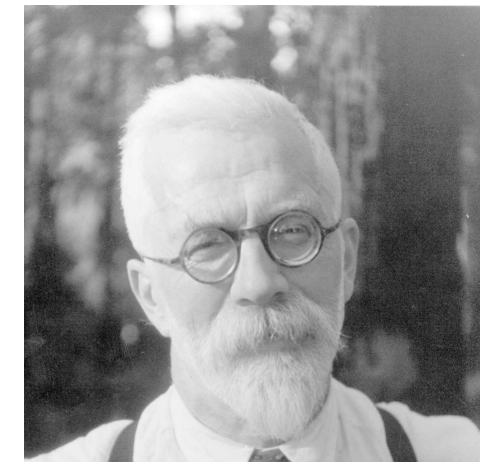


Vektorquantisierung

0,5 Bits / Pixel

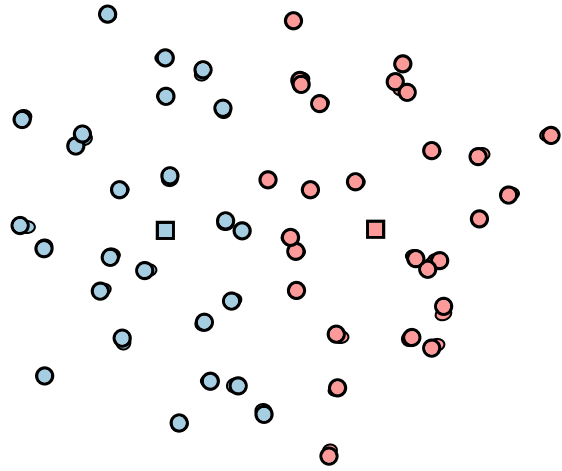


1,9 Bits / Pixel

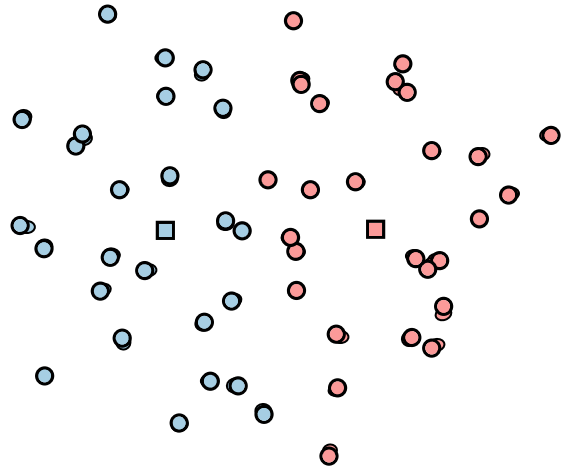


8 Bits / Pixel

k -MEANS: Probleme

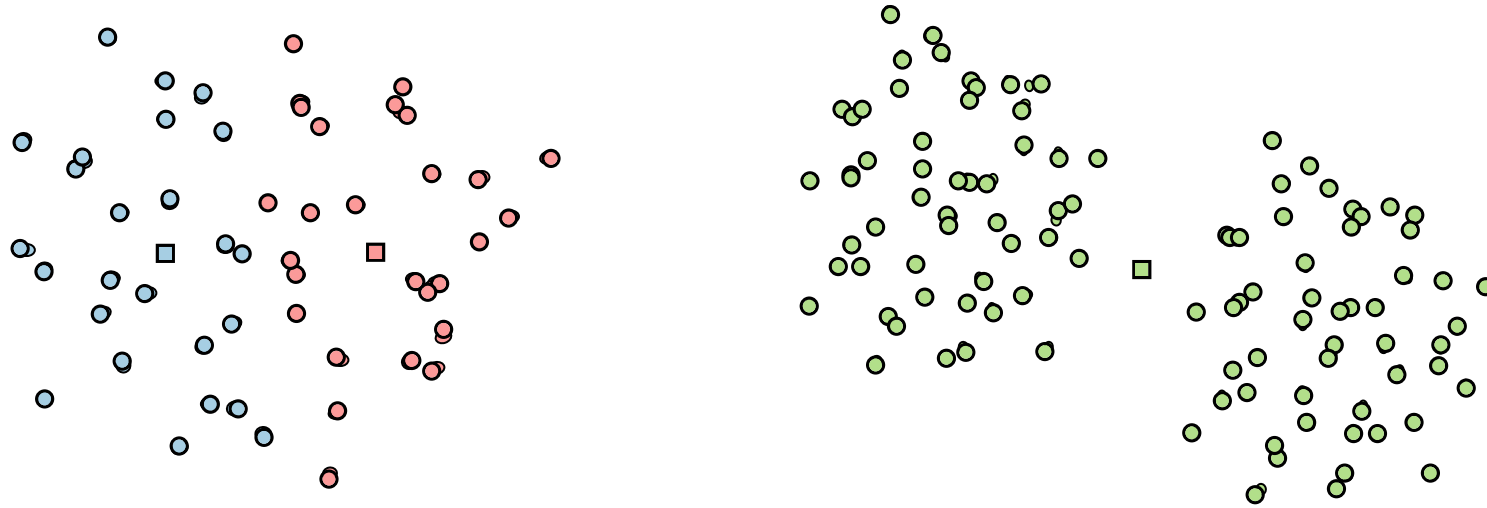


k -MEANS: Probleme



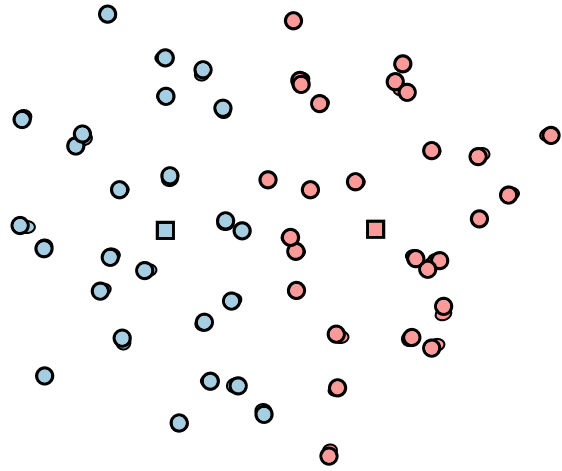
1 Cluster wäre besser

k -MEANS: Probleme

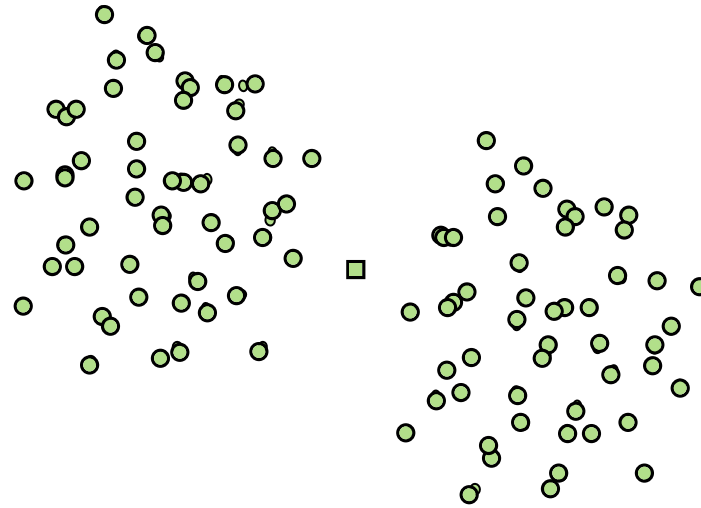


1 Cluster wäre besser

k -MEANS: Probleme

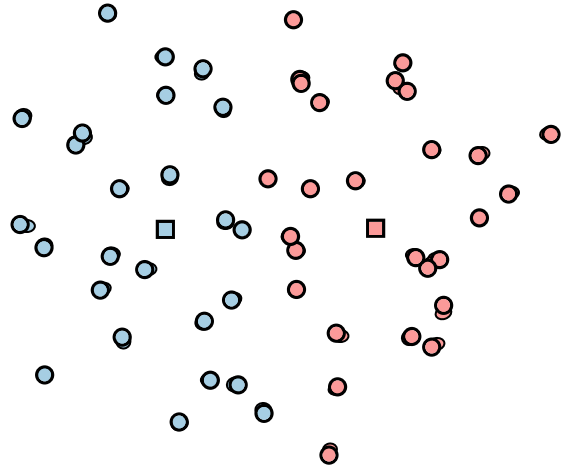


1 Cluster wäre besser

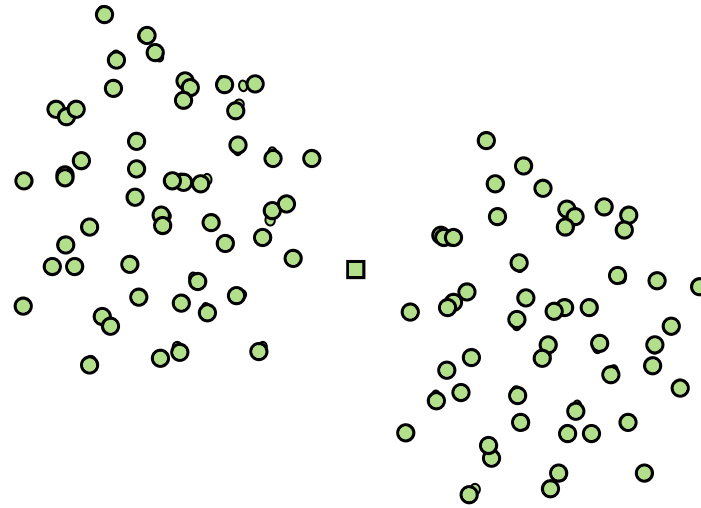


2 Cluster wären besser

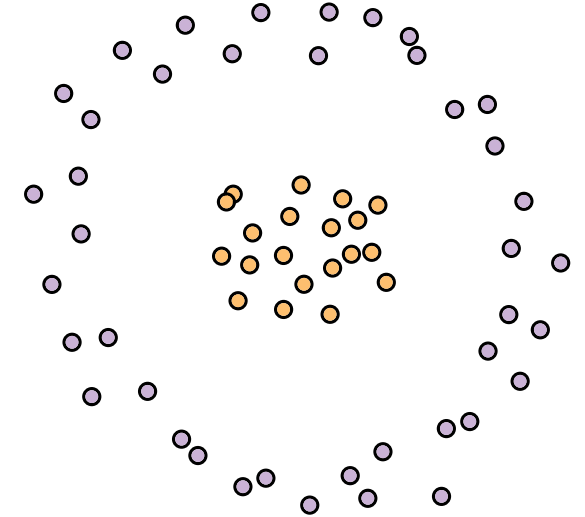
k -MEANS: Probleme



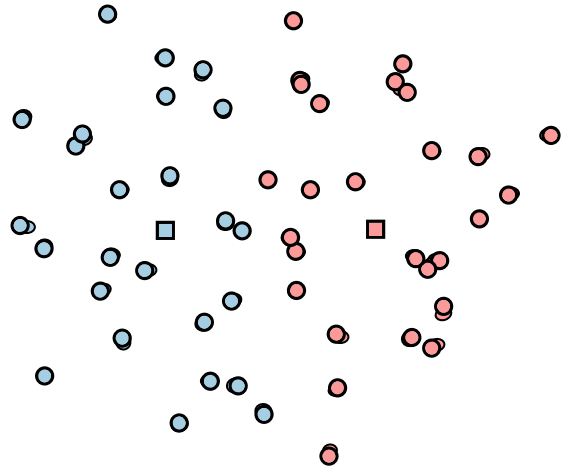
1 Cluster wäre besser



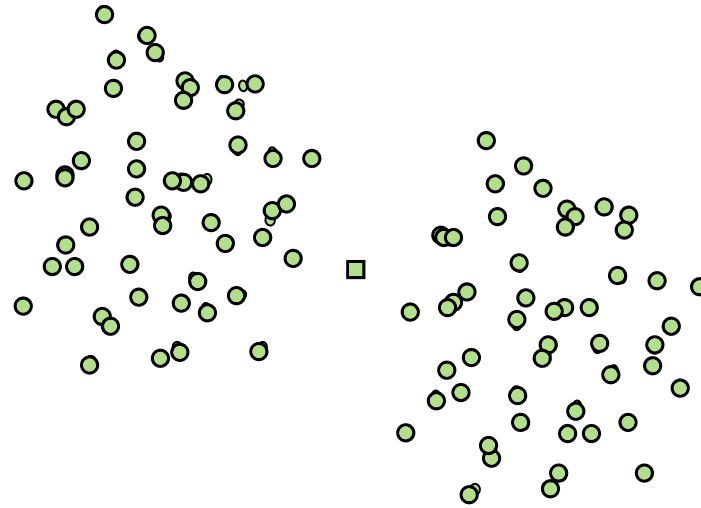
2 Cluster wären besser



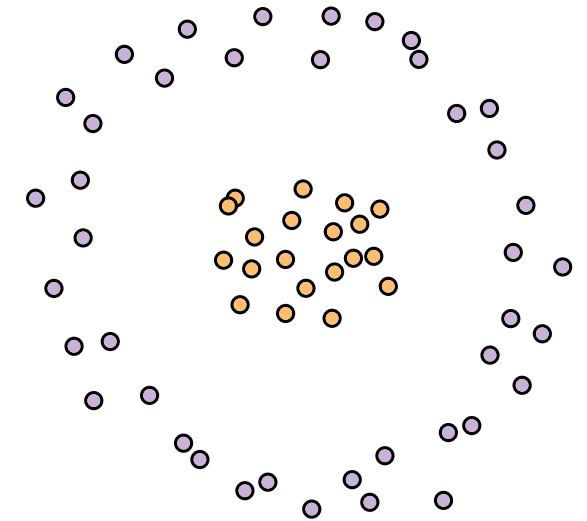
k -MEANS: Probleme



1 Cluster wäre besser

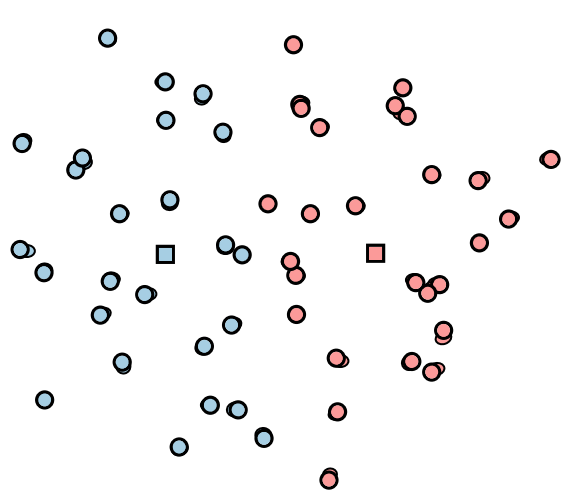


2 Cluster wären besser

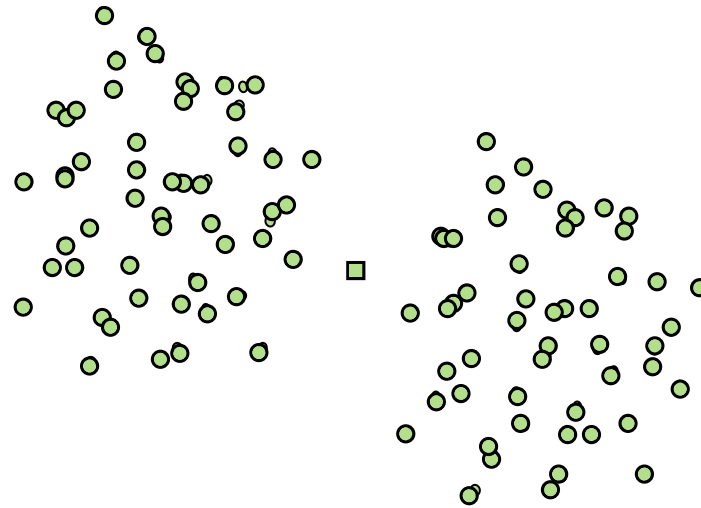


kann k -Means nicht lösen

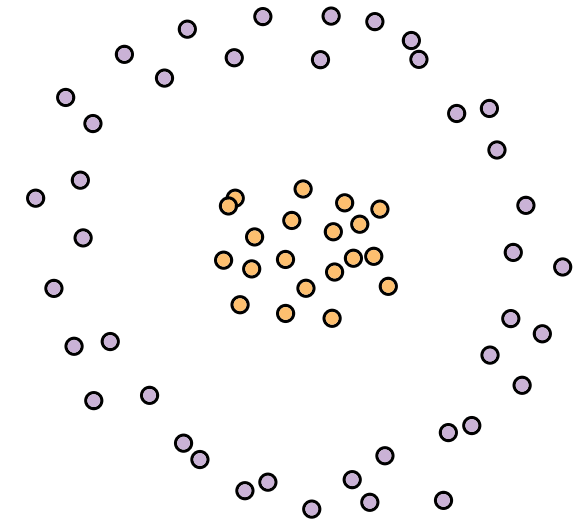
k -MEANS: Probleme



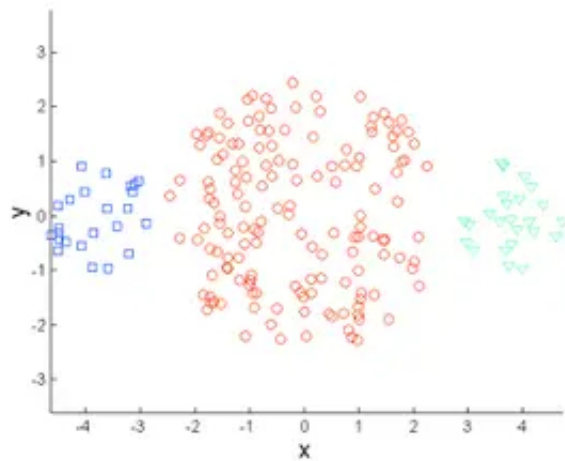
1 Cluster wäre besser



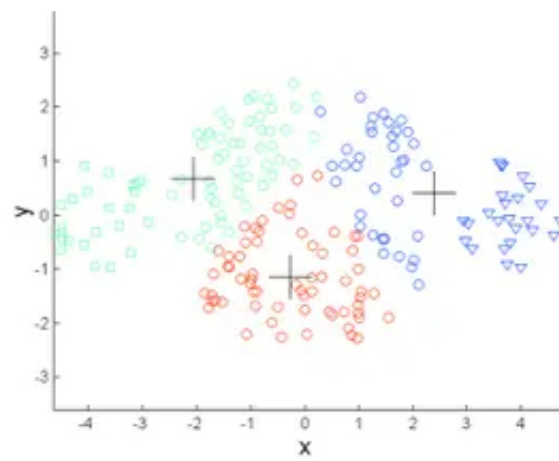
2 Cluster wären besser



kann k -Means nicht lösen



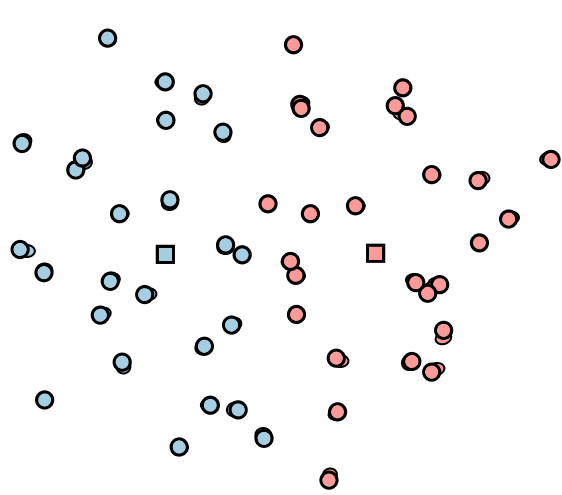
Original Points



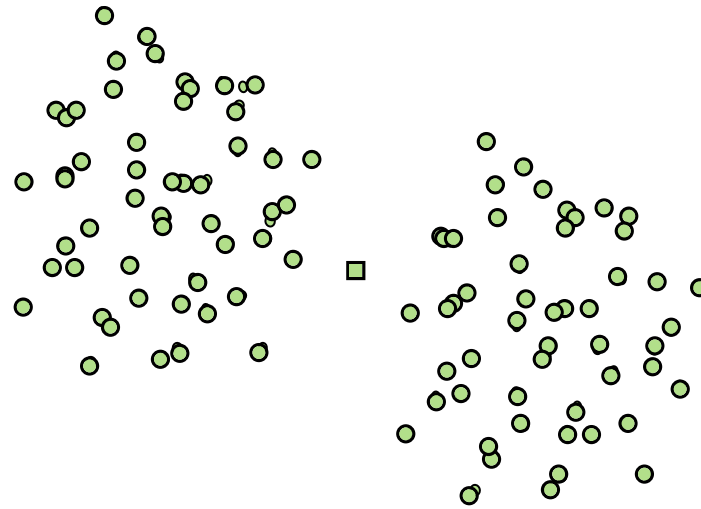
K-means (k = 3)

Unterschiedliche Clustergrößen

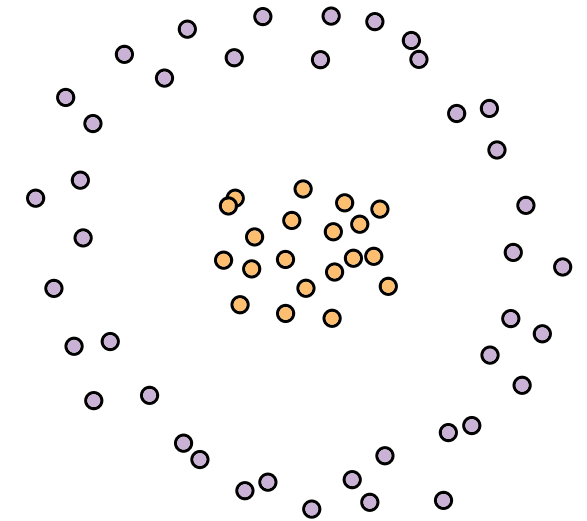
k -MEANS: Probleme



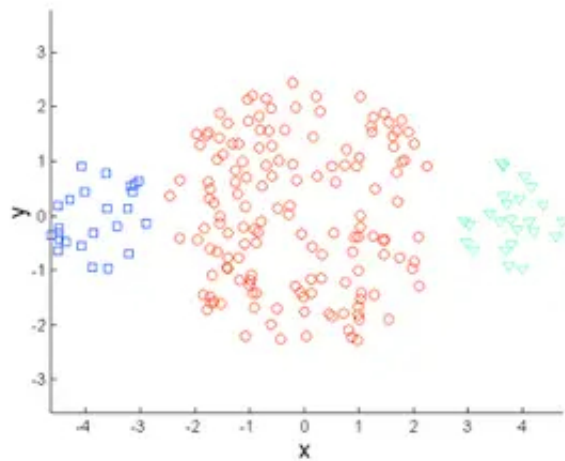
1 Cluster wäre besser



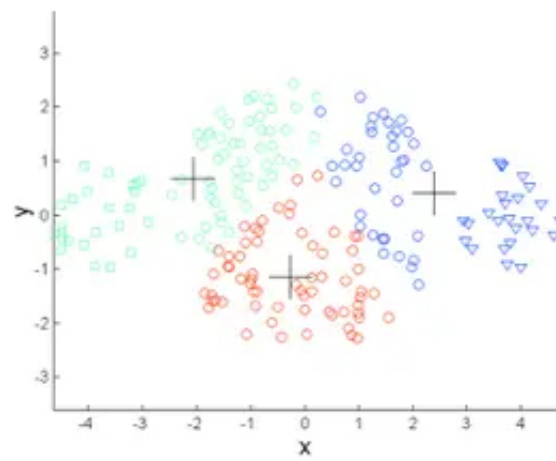
2 Cluster wären besser



kann k -Means nicht lösen

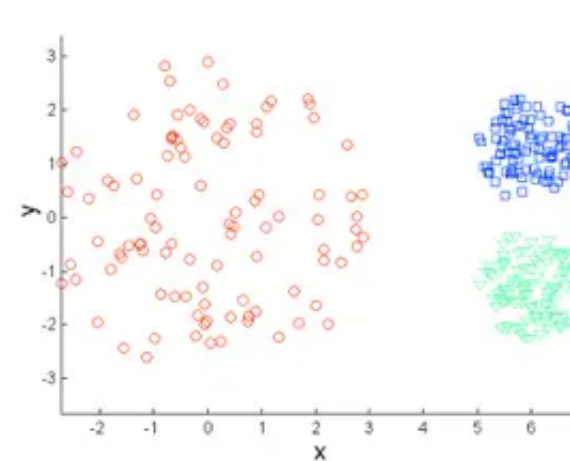


Original Points

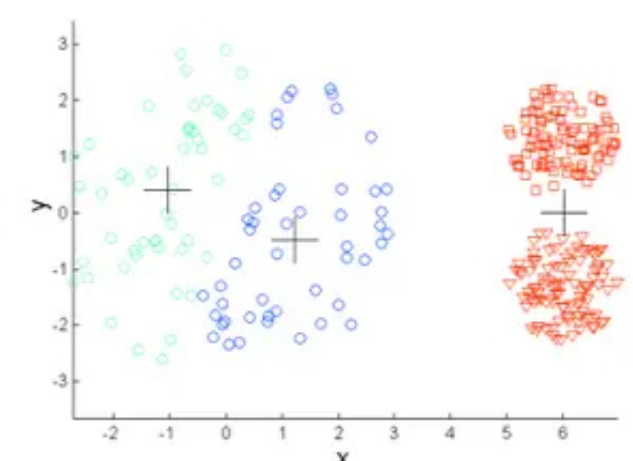


K-means ($k = 3$)

Unterschiedliche Clustergrößen



Original Points



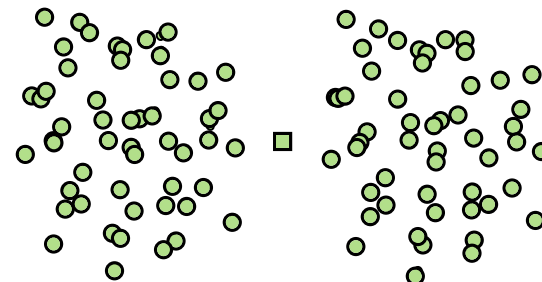
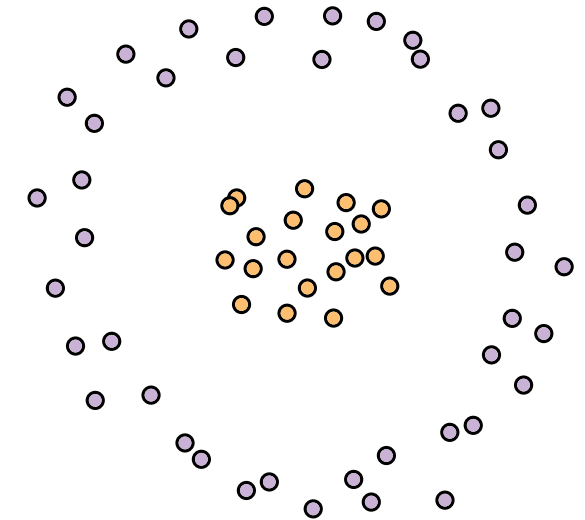
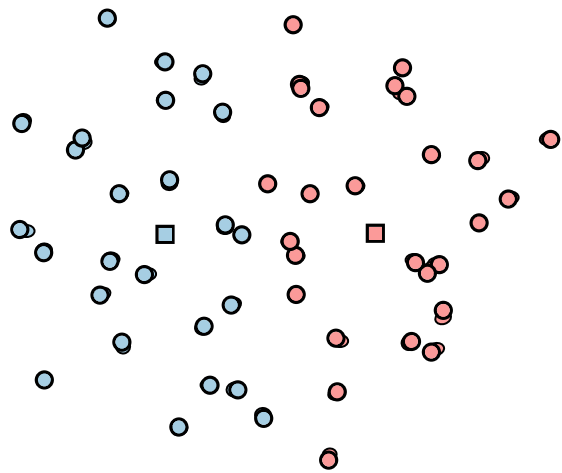
K-means ($k = 3$)

Unterschiedliche Punktdichte

Algorithmen für geographische Informationssysteme

13. Vorlesung Clustering

Teil III: DBSCAN



DBSCAN



Gegeben.

Menge X von (Daten)punkten

DBSCAN



Gegeben.

Menge X von (Daten)punkten, Distanzfunktion

DBSCAN



Gegeben.

Menge X von (Daten)punkten, Distanzfunktion
 $d(\cdot, \cdot)$,

DBSCAN



Gegeben.

Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε

DBSCAN



Gegeben.

Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

DBSCAN



Gegeben.

Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

DBSCAN



Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

DBSCAN



Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

DBSCAN



Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

Ein Punkt $p \in P$ ist **direkt dichte-erreichbar** von einem Punkt q wenn:

Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

Ein Punkt $p \in P$ ist **direkt dichte-erreichbar** von einem Punkt q wenn:

- $p \in N_\varepsilon(q)$

Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

Ein Punkt $p \in P$ ist **direkt dichte-erreichbar** von einem Punkt q wenn:

- $p \in N_\varepsilon(q)$
- $|N_\varepsilon(q)| \geq k$ (q ist ein Kern)

Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

Ein Punkt $p \in P$ ist **direkt dichte-erreichbar** von einem Punkt q wenn:

- $p \in N_\varepsilon(q)$
- $|N_\varepsilon(q)| \geq k$ (q ist ein Kern)

nicht symmetrisch!

Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

Ein Punkt $p \in P$ ist **direkt dichte-erreichbar** von einem Punkt q wenn:

- $p \in N_\varepsilon(q)$
- $|N_\varepsilon(q)| \geq k$ (q ist ein Kern)

nicht symmetrisch!

Ein Punkt $p \in P$ ist **dichte-erreichbar** von einem Punkt q

$\Leftrightarrow$ es gibt einen Pfad von direkt dichte-erreichbaren Punkten von q nach p .

Gegeben. Menge X von (Daten)punkten, Distanzfunktion $d(\cdot, \cdot)$, max. Distanz ε und min. Anzahl k

Die **ε -Nachbarschaft** eines Punkts $p \in X$ ist $N_\varepsilon(p) = \{ q \in X \mid d(p, q) \leq \varepsilon \}$.

Ein Punkt $p \in P$ heißt **Kern(punkt)** wenn $|N_\varepsilon(p)| \geq k$.

Ein Punkt $p \in P$ ist **direkt dichte-erreichbar** von einem Punkt q wenn:

- $p \in N_\varepsilon(q)$
- $|N_\varepsilon(q)| \geq k$ (q ist ein Kern)

nicht symmetrisch!

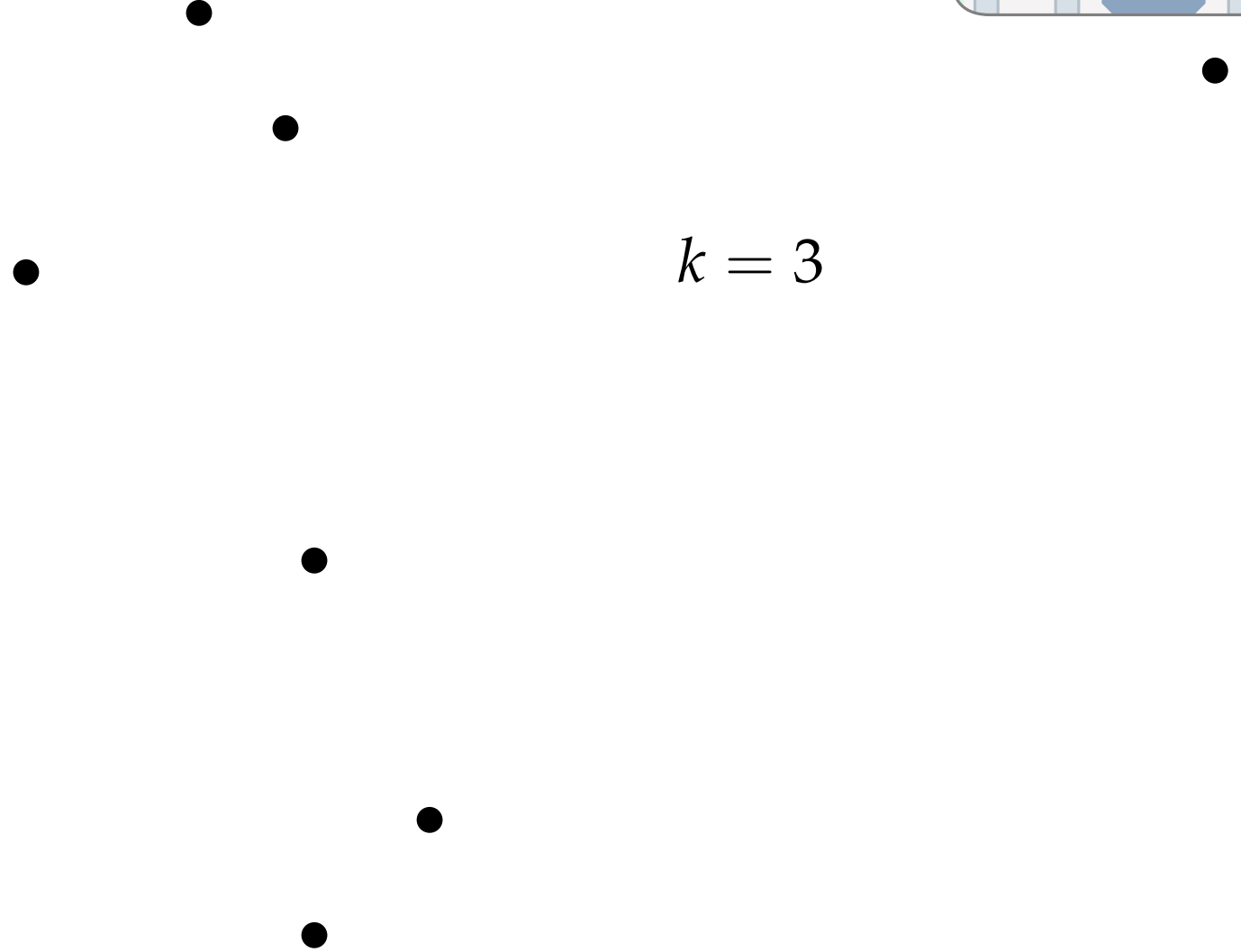
Ein Punkt $p \in P$ ist **dichte-erreichbar** von einem Punkt q

$\Leftrightarrow$ es gibt einen Pfad von direkt dichte-erreichbaren Punkten von q nach p .

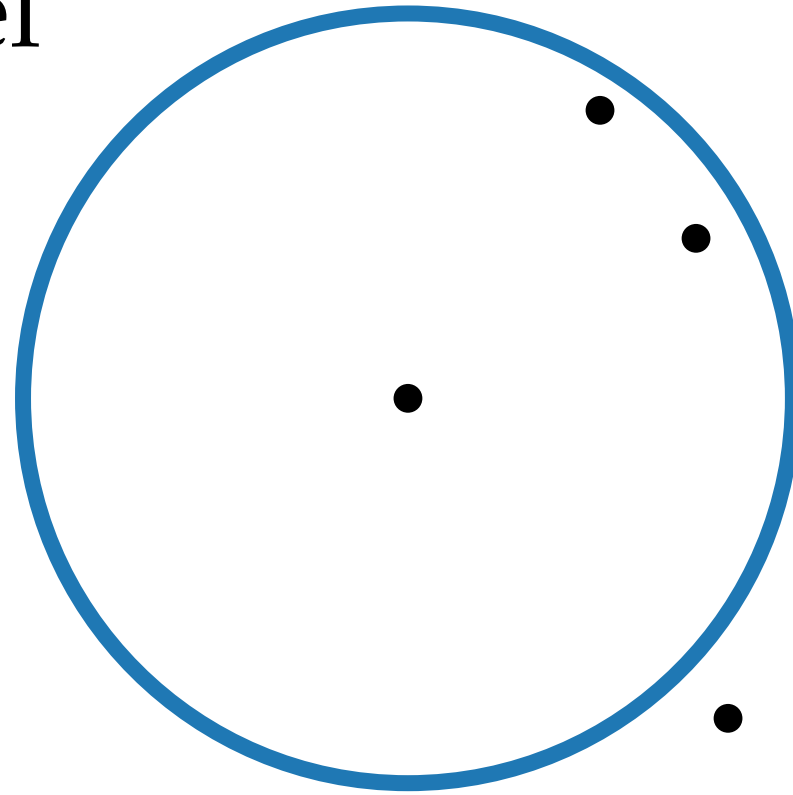
Zwei Punkte $p, q \in P$ sind **dichte-verbunden**

$\Leftrightarrow$ es gibt einen Kern r , so dass sowohl p als auch q von r dichte-erreichbar sind.

DBSCAN: Beispiel



DBSCAN: Beispiel

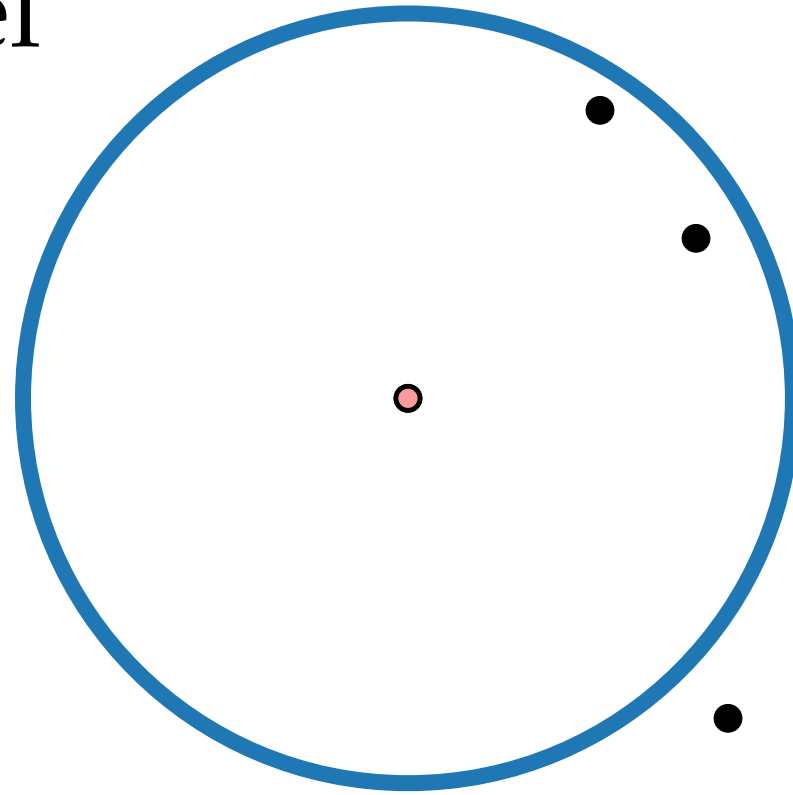


$k = 3$

Distanz ε



DBSCAN: Beispiel



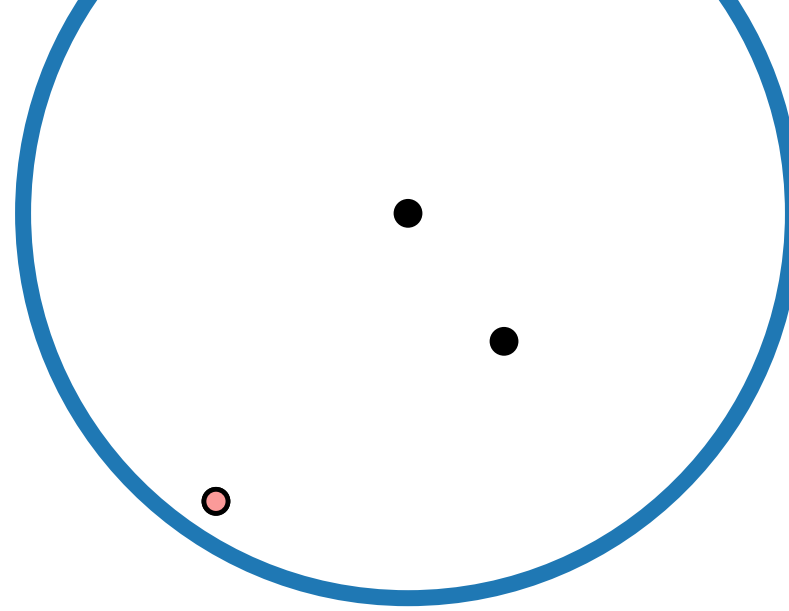
$k = 3$

Distanz ϵ

Kerne



DBSCAN: Beispiel



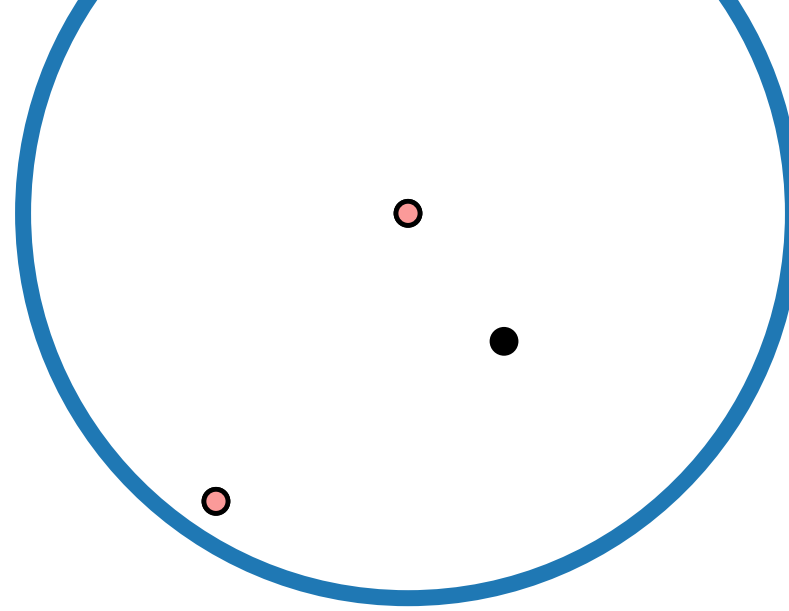
$k = 3$

Distanz ε

Kerne



DBSCAN: Beispiel



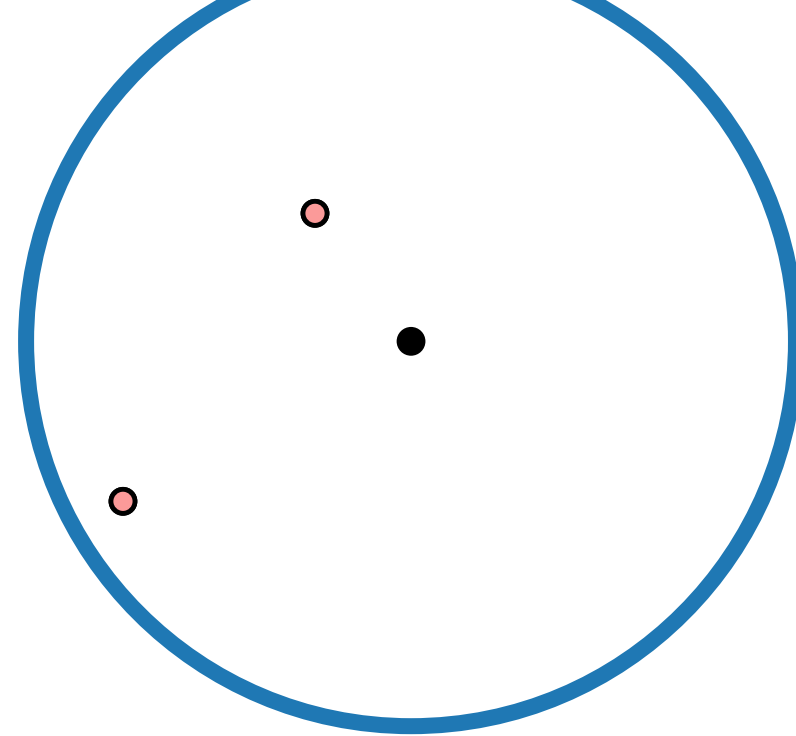
$k = 3$

Distanz ε

Kerne



DBSCAN: Beispiel



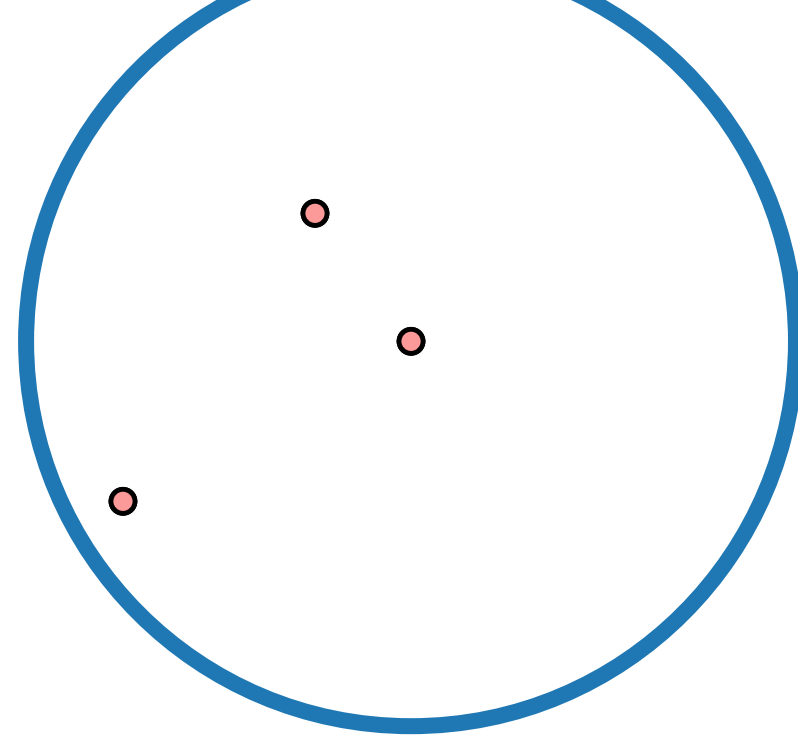
$k = 3$

Distanz ϵ

Kerne



DBSCAN: Beispiel



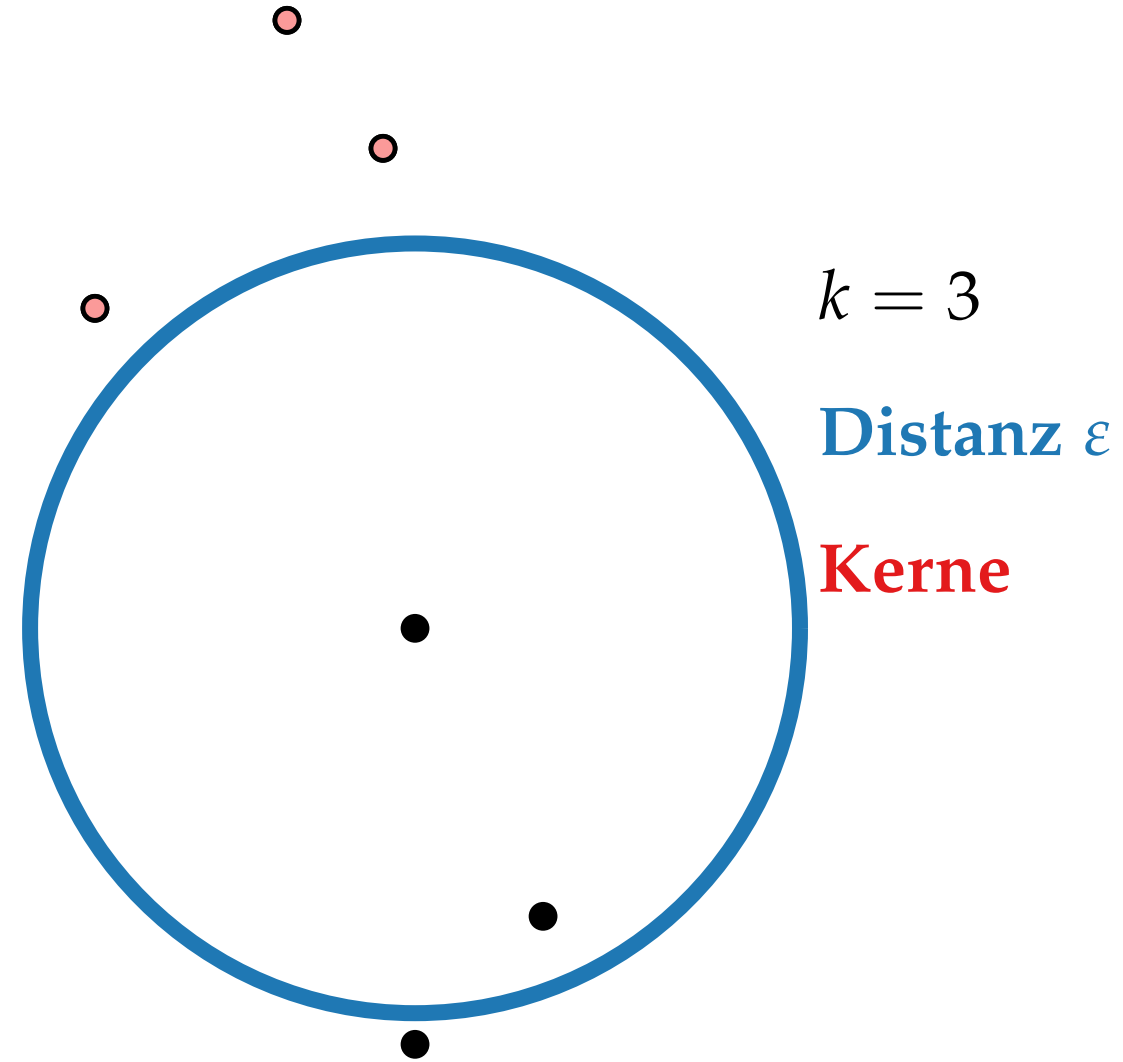
$k = 3$

Distanz ϵ

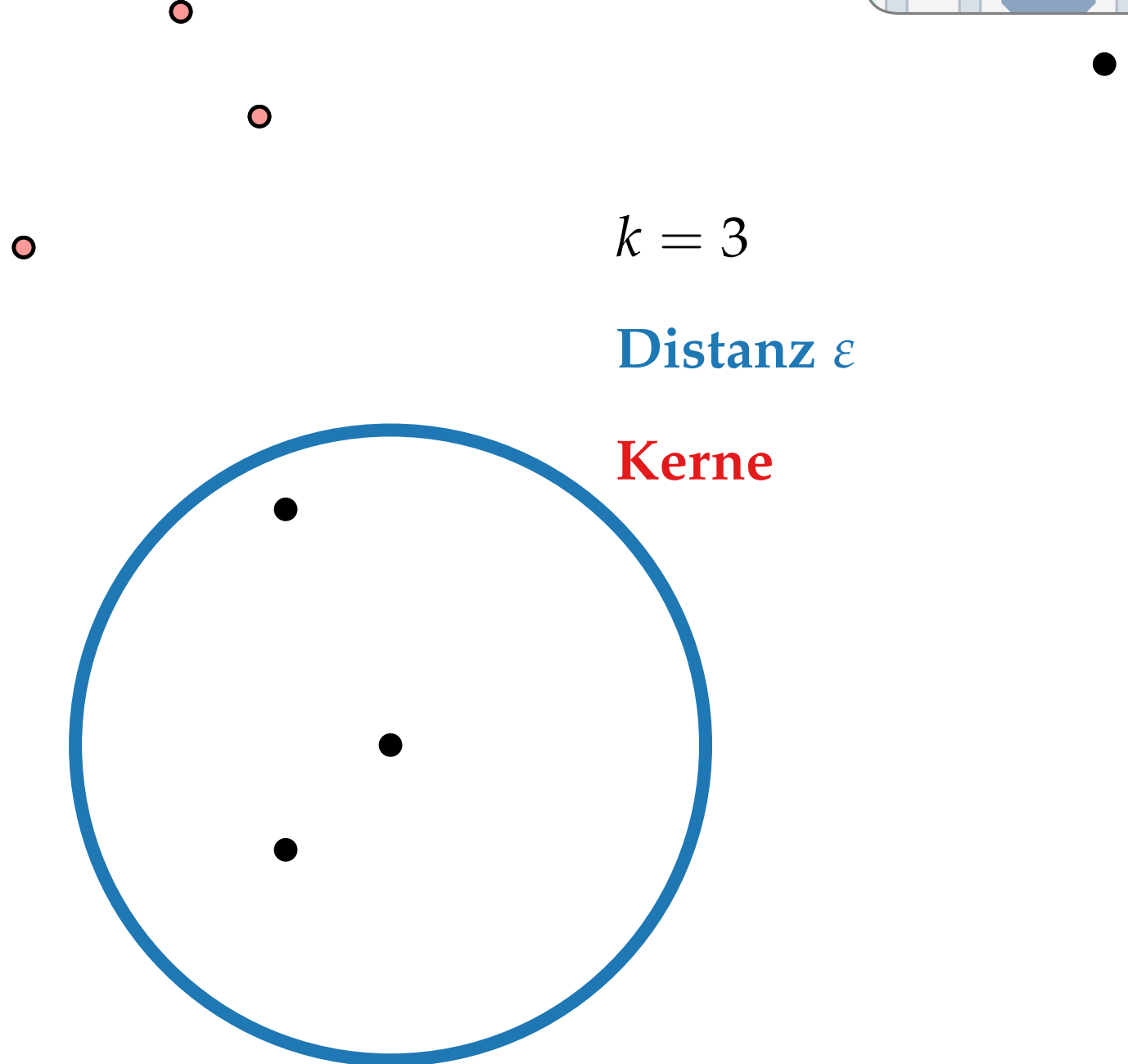
Kerne



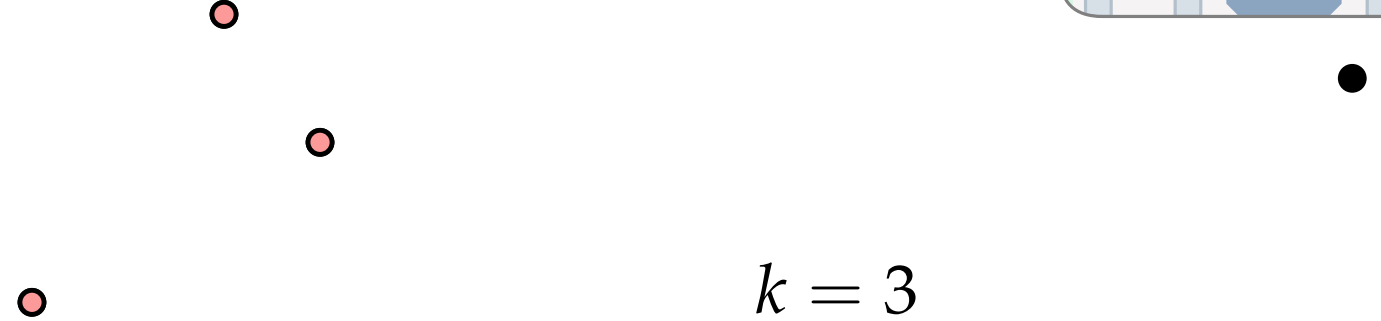
DBSCAN: Beispiel



DBSCAN: Beispiel



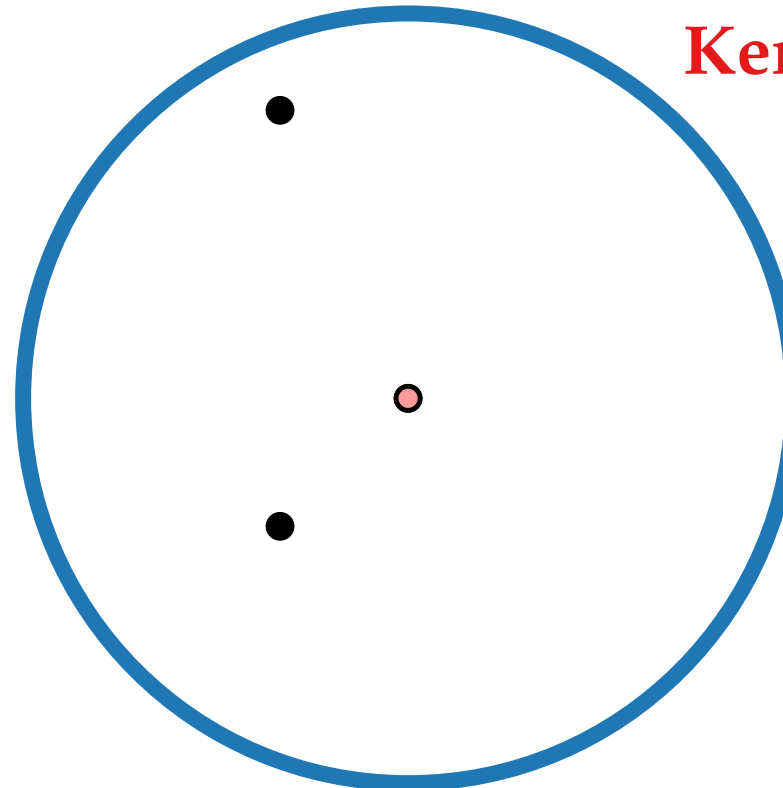
DBSCAN: Beispiel



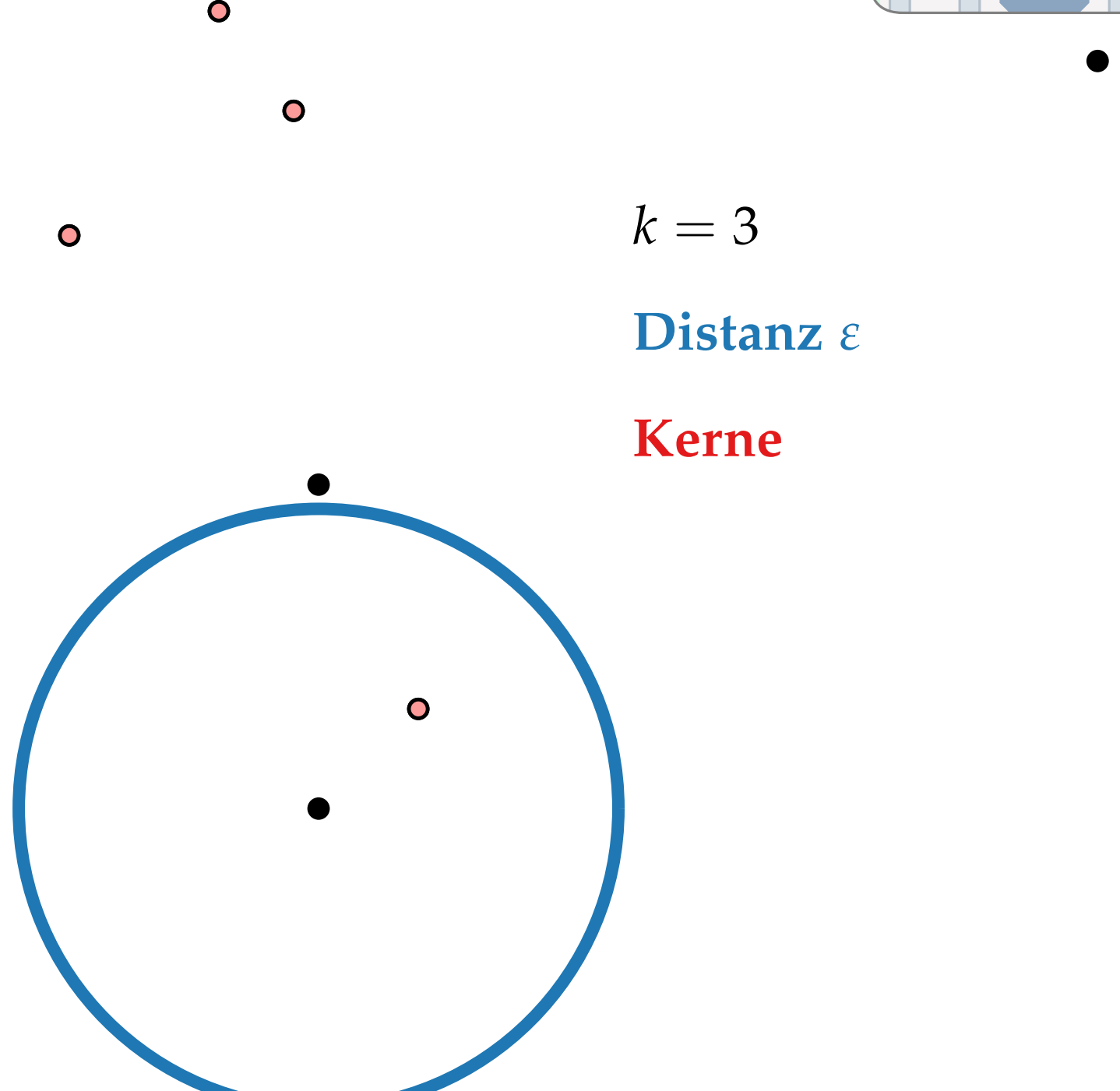
$k = 3$

Distanz ε

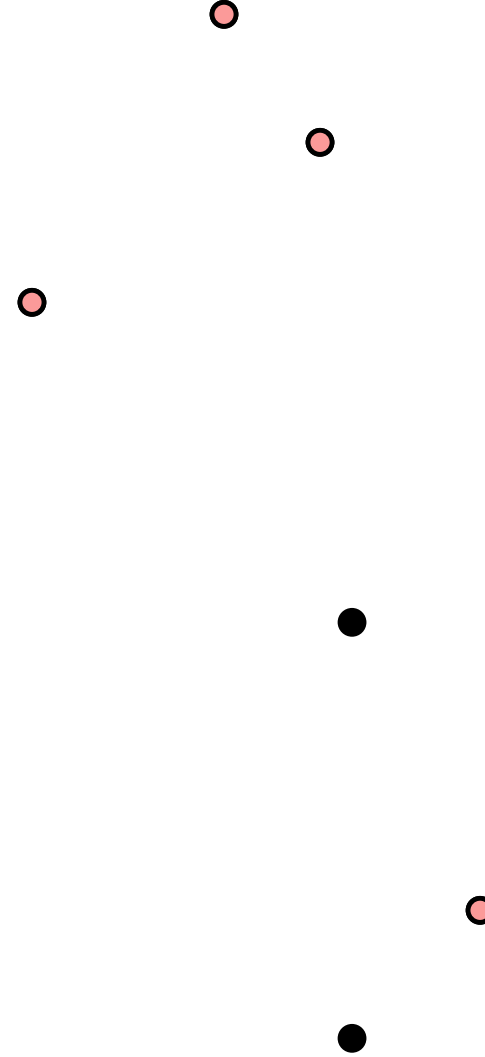
Kerne



DBSCAN: Beispiel



DBSCAN: Beispiel



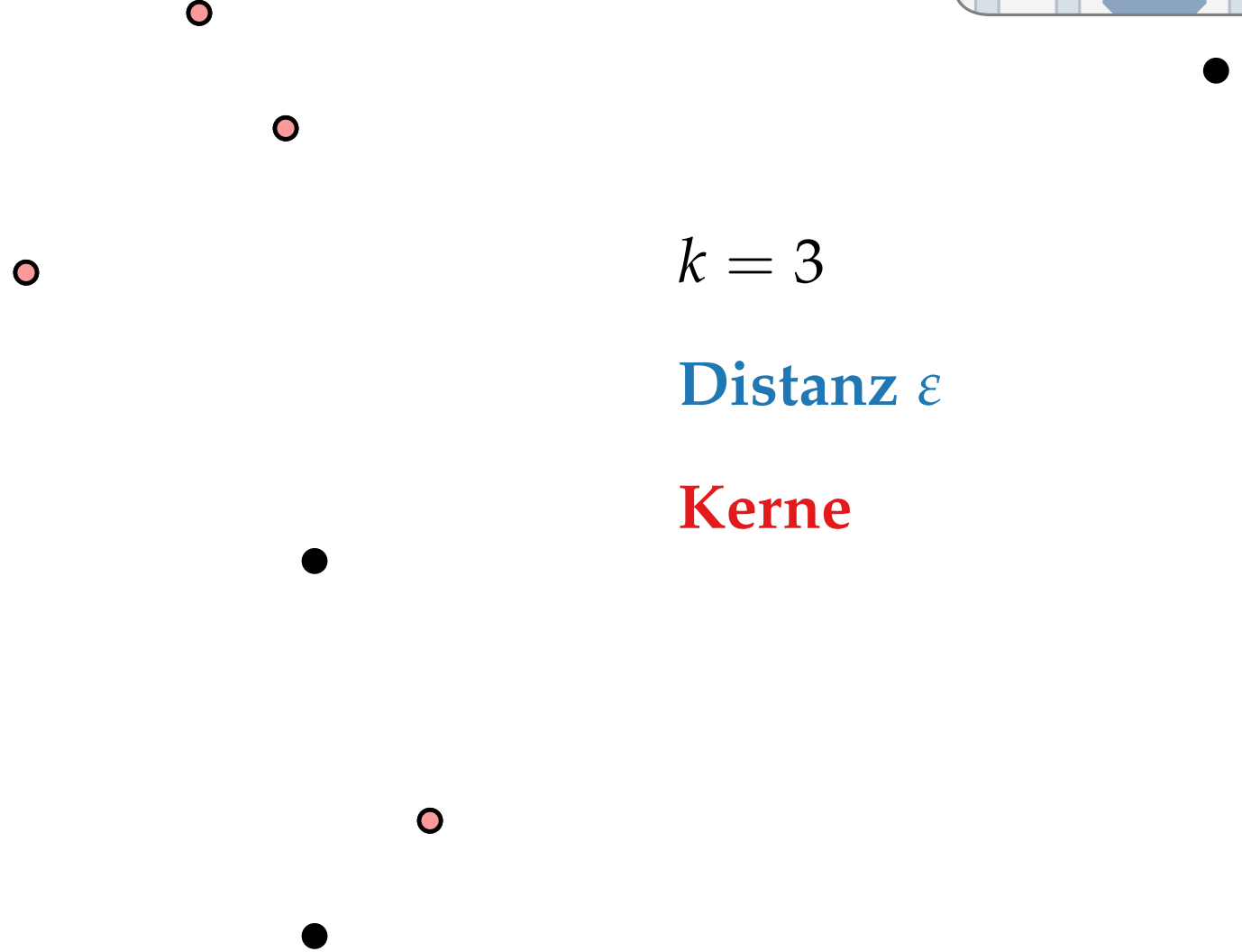
$k = 3$

Distanz ε

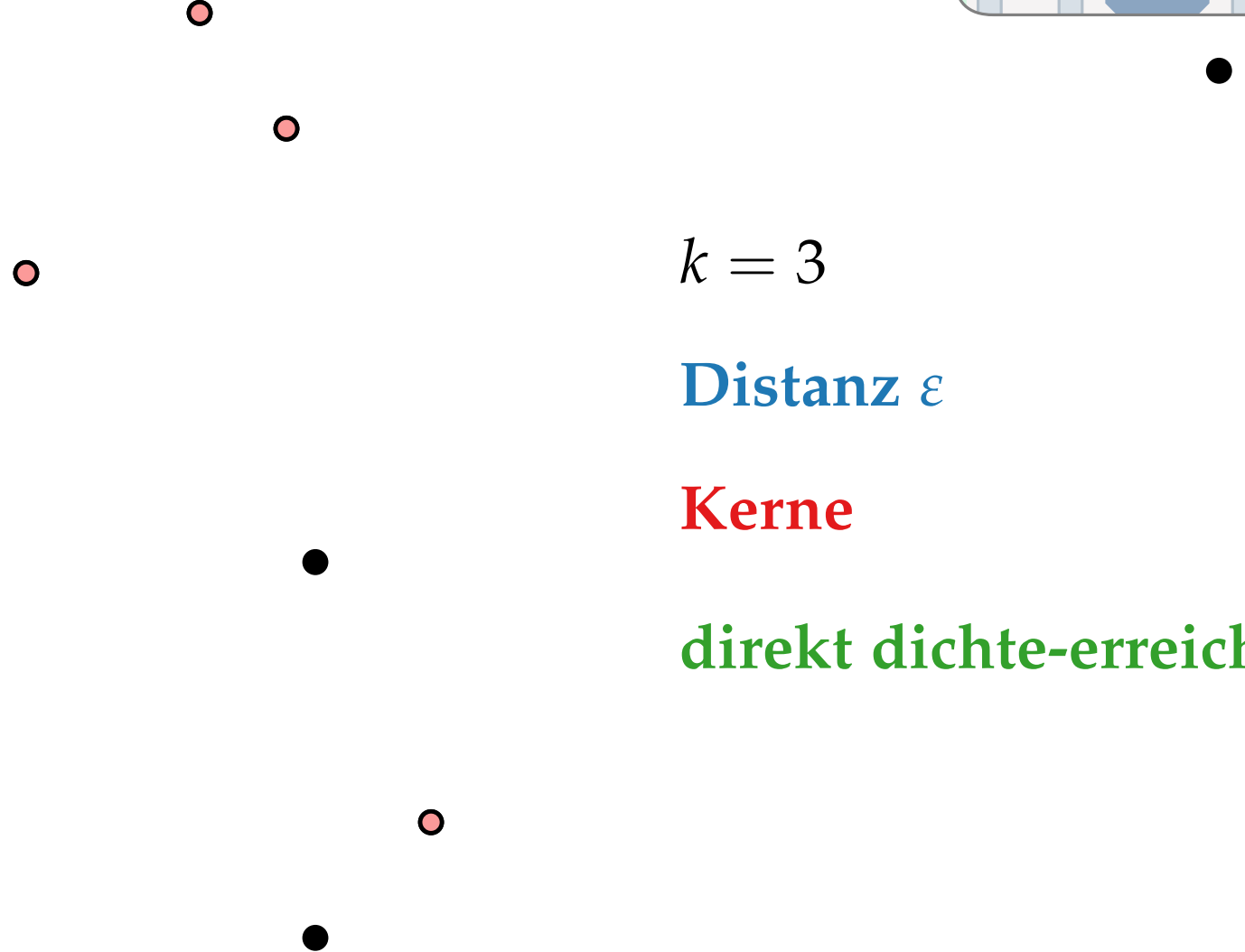
Kerne



DBSCAN: Beispiel



DBSCAN: Beispiel



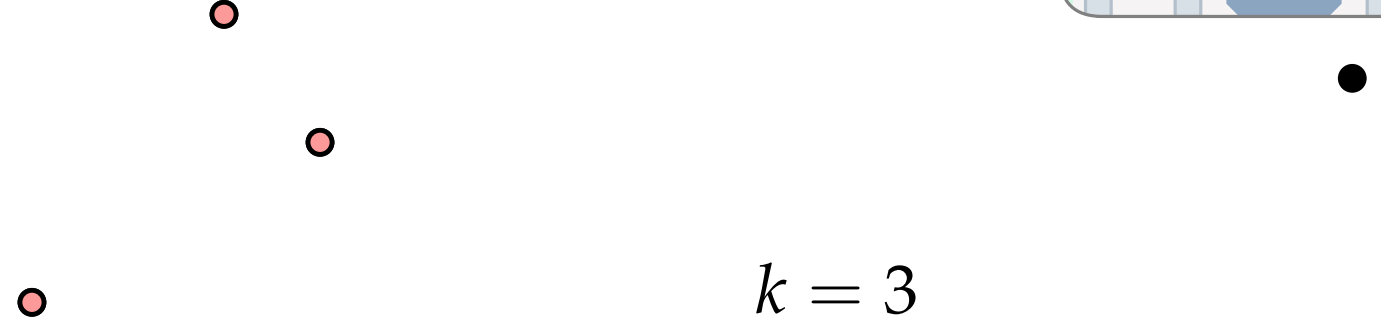
$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar

DBSCAN: Beispiel

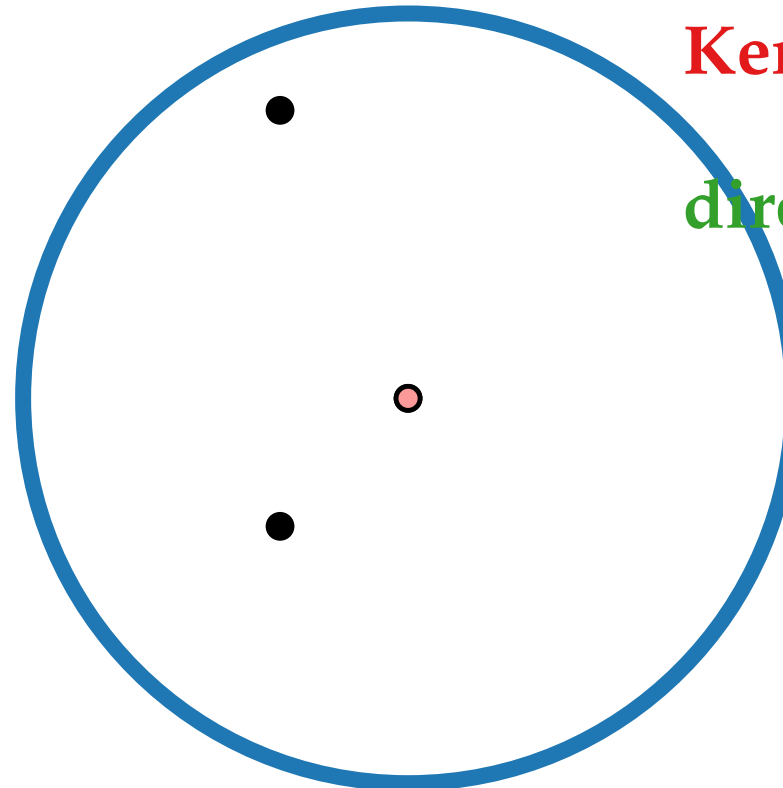


$k = 3$

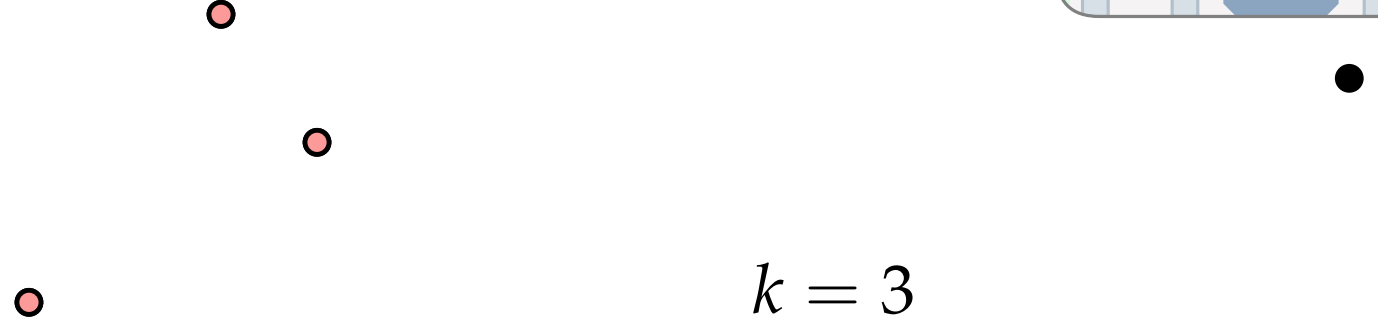
Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

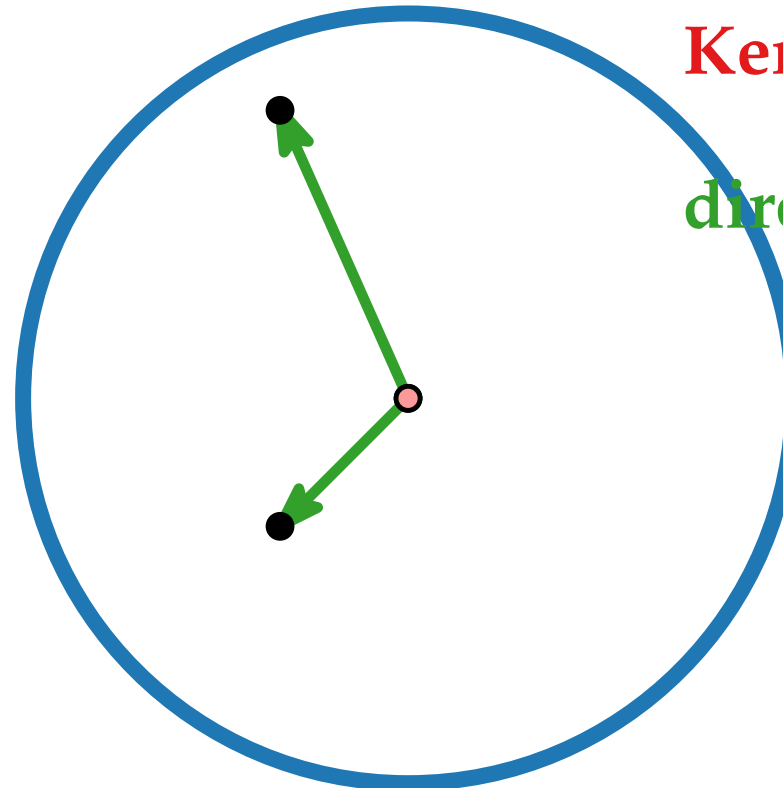


$k = 3$

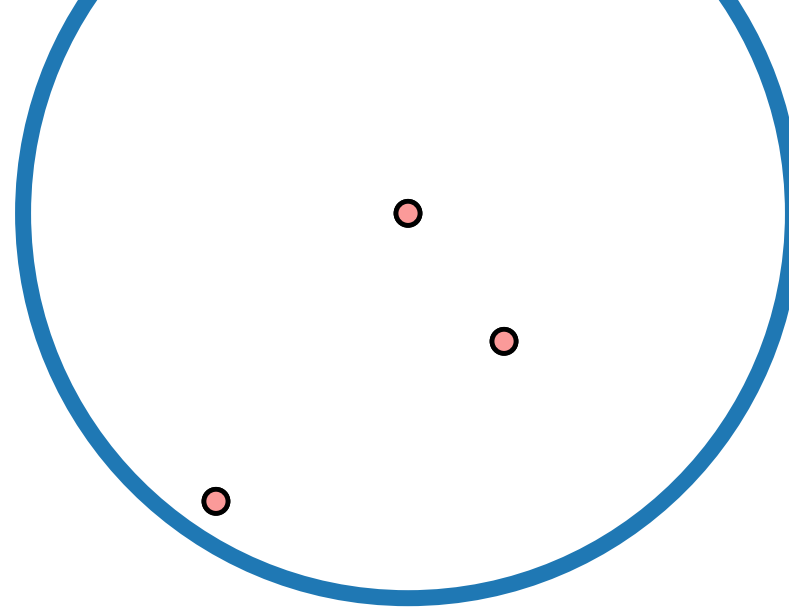
Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

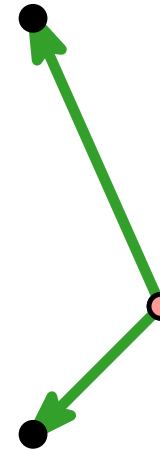


$k = 3$

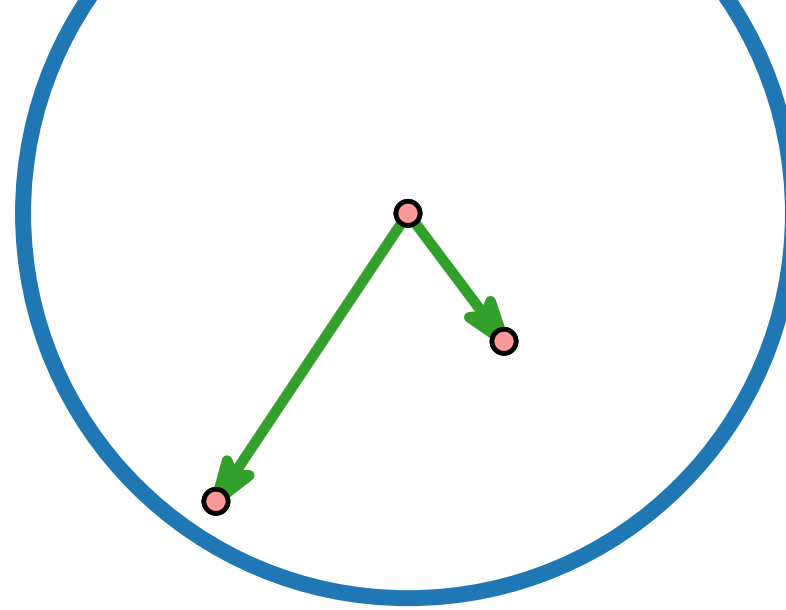
Distanz ϵ

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

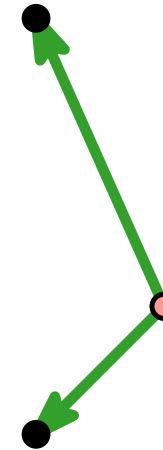


$k = 3$

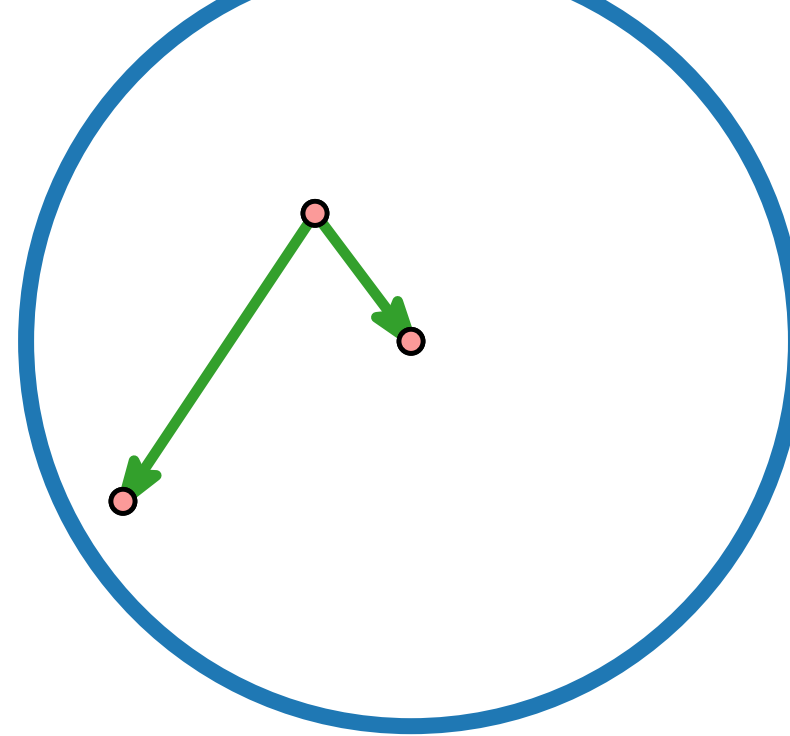
Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

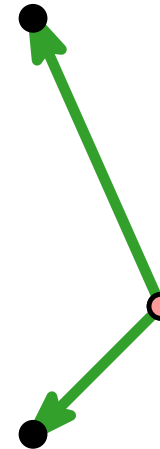


$k = 3$

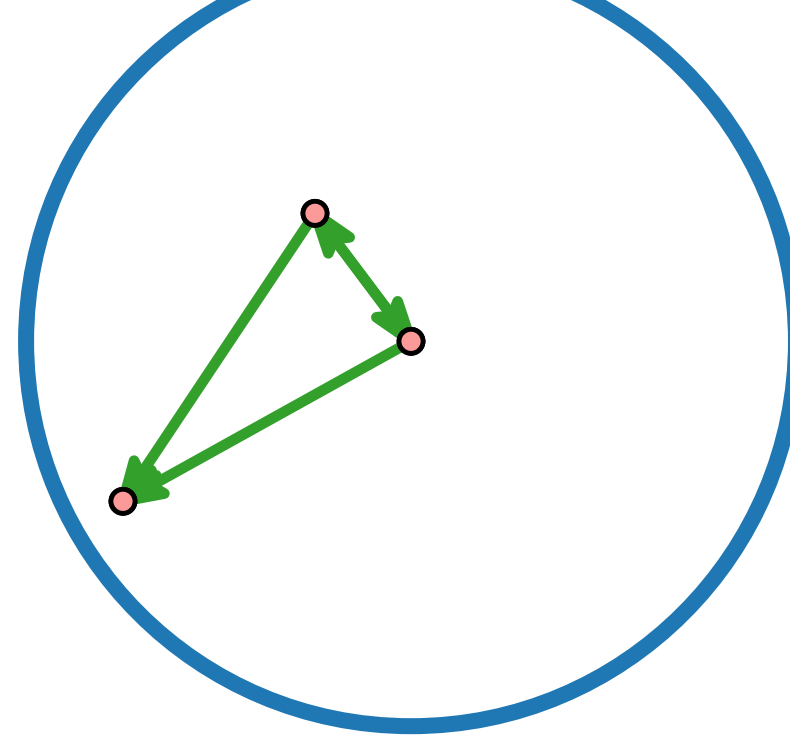
Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

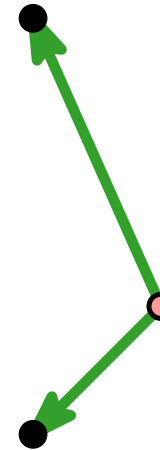


$k = 3$

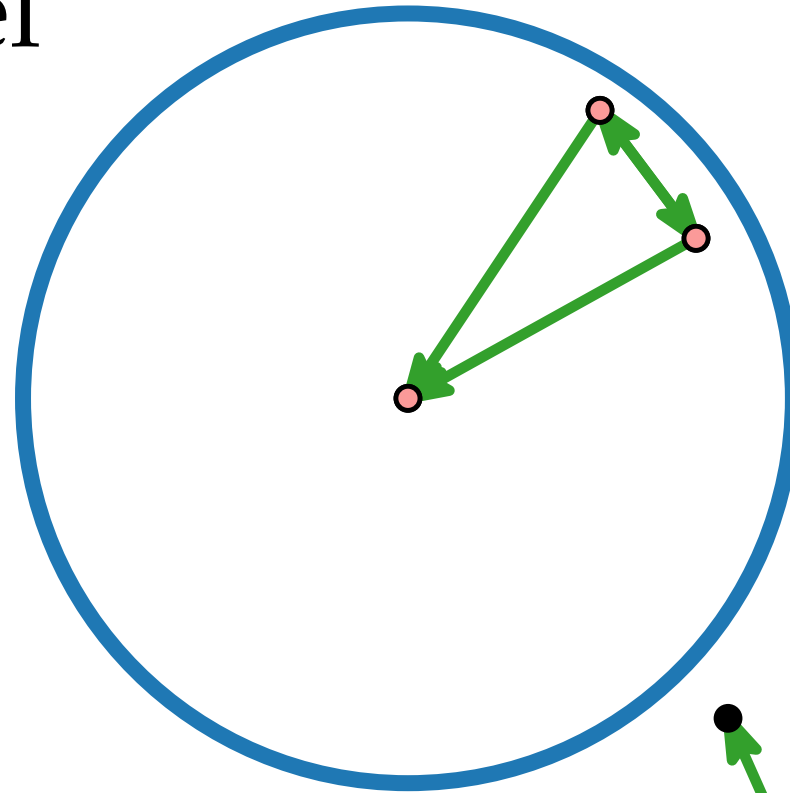
Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

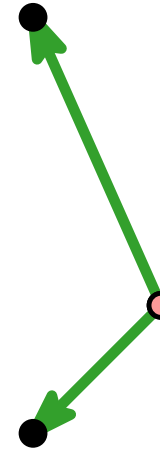


$k = 3$

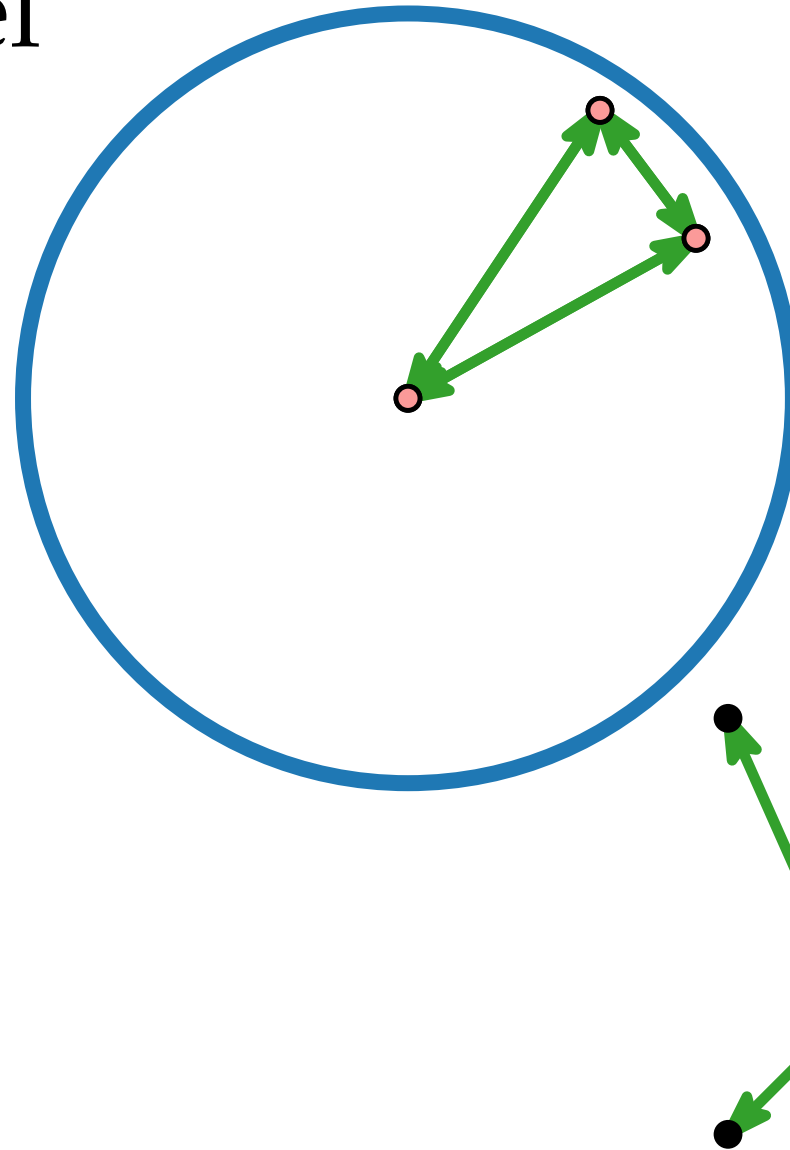
Distanz ϵ

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel



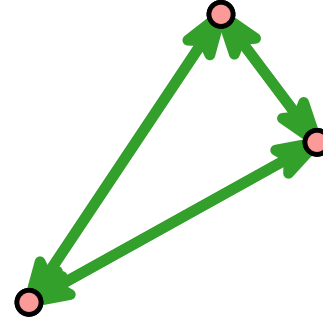
$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar

DBSCAN: Beispiel

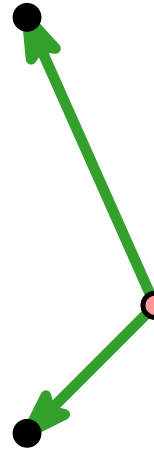


$k = 3$

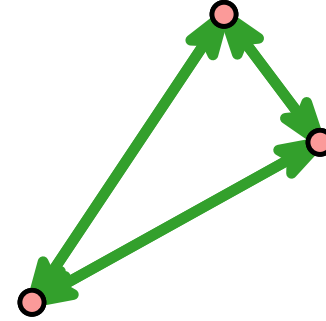
Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN: Beispiel

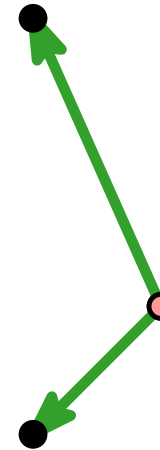


$k = 3$

Distanz ε

Kerne

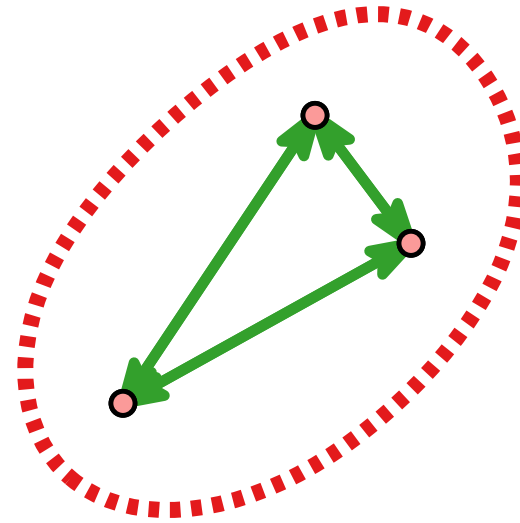
direkt dichte-erreichbar



DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN: Beispiel

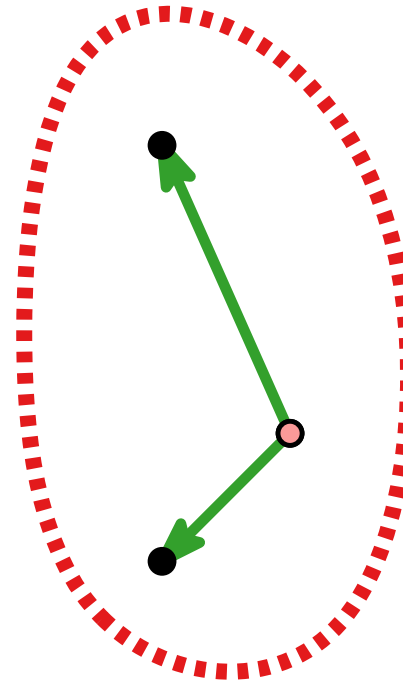


$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar

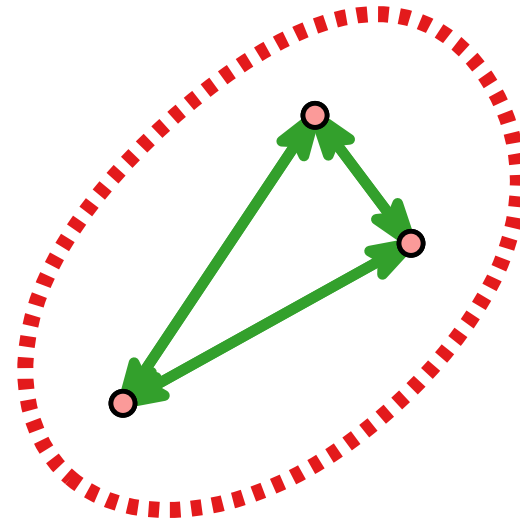


DBSCAN Clustering

DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN: Beispiel



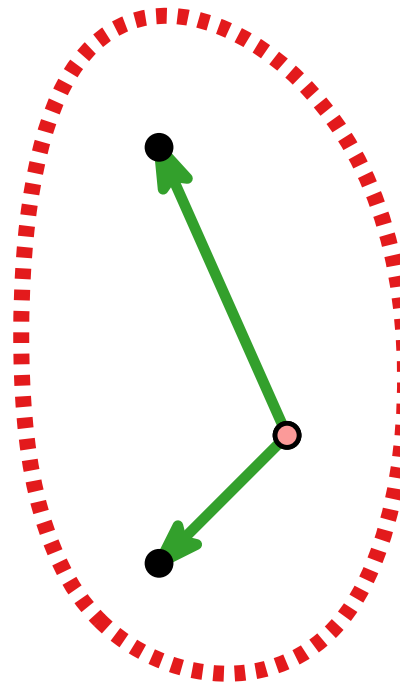
Rauschen / Ausreißer

$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar

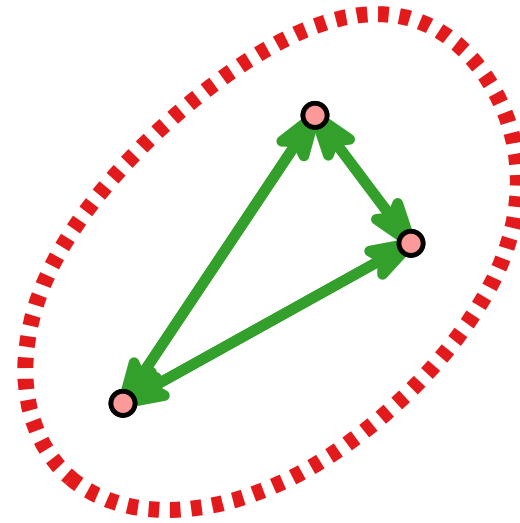


DBSCAN Clustering

DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN: Beispiel



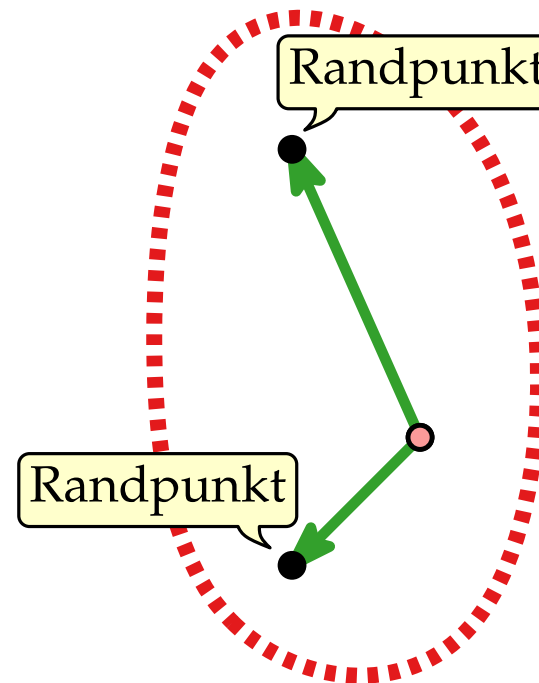
Rauschen / Ausreißer

$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar

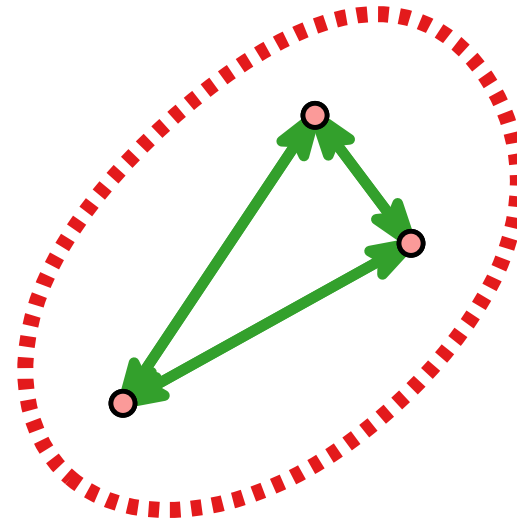


DBSCAN Clustering

DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN: Beispiel



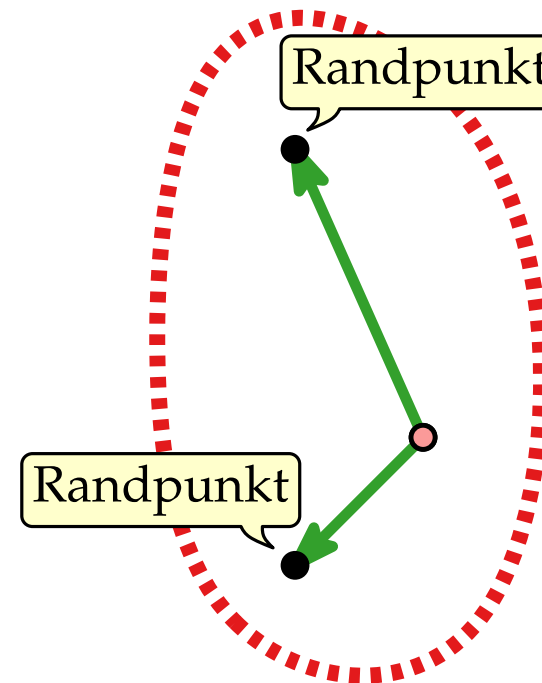
Rauschen / Ausreißer

$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN Clustering

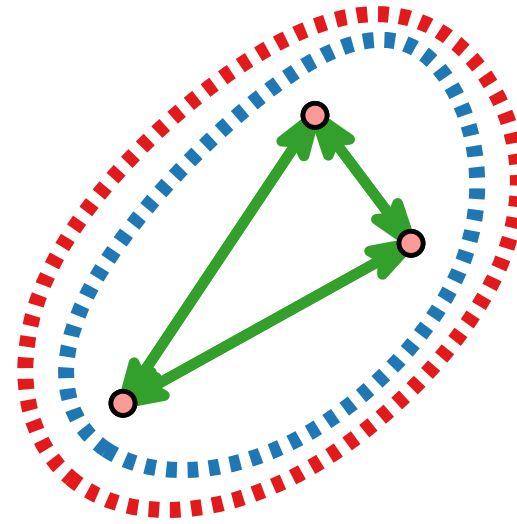
DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN*:

(und Kerne)

DBSCAN: Beispiel



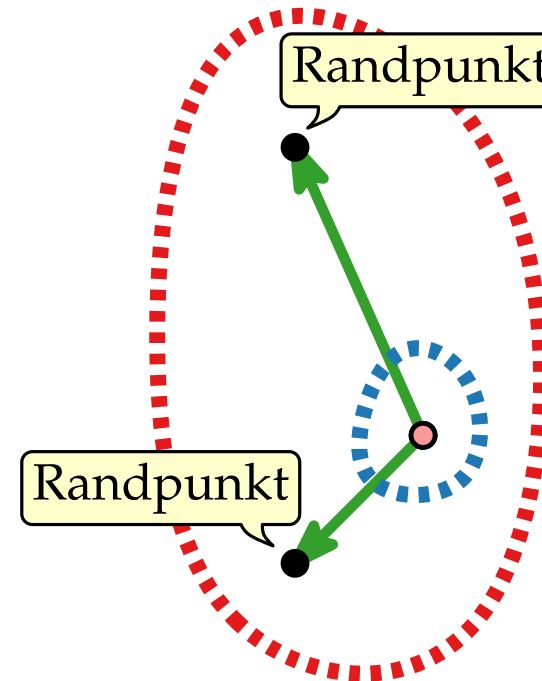
Rauschen / Ausreißer

$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar



DBSCAN Clustering

DBSCAN* Clustering

DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN*:

(und Kerne)

DBSCAN: Beispiel

Laufzeit.

Naiver Algorithmus läuft in $\mathcal{O}(n^2)$ Zeit.



Rauschen / Ausreißer

$k = 3$

Distanz ε

Kerne

direkt dichte-erreichbar

DBSCAN Clustering

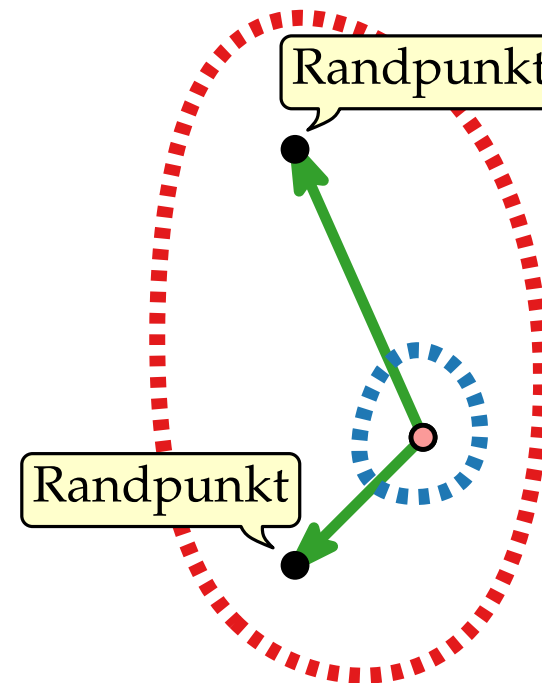
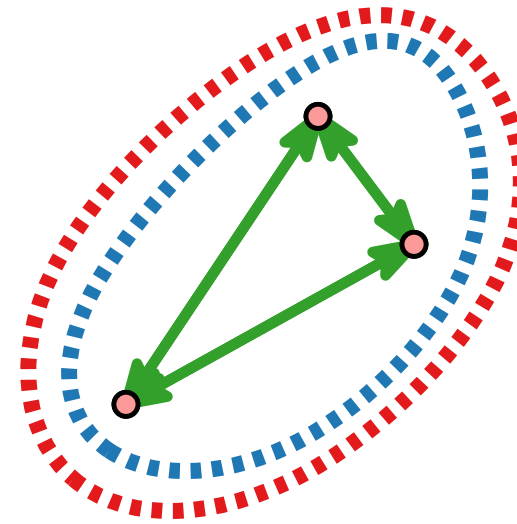
DBSCAN* Clustering

DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden

DBSCAN*:

(und Kerne)



DBSCAN: Beispiel

Laufzeit.

Naiver Algorithmus läuft in $\mathcal{O}(n^2)$ Zeit.

“Since the Eps-neighborhoods are expected to be small compared to the size of the whole data space, the average run time complexity of a single region query is $\mathcal{O}(\log n)$. [...] Thus, the average run time complexity of DBSCAN is $\mathcal{O}(n \log n)$.”

DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow p$ und q sind dichte-verbunden



Rauschen / Ausreißer

$k = 3$

Distanz ε

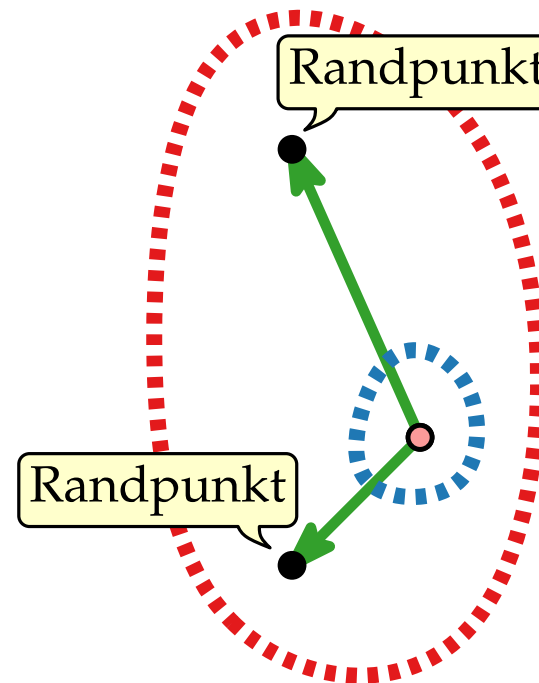
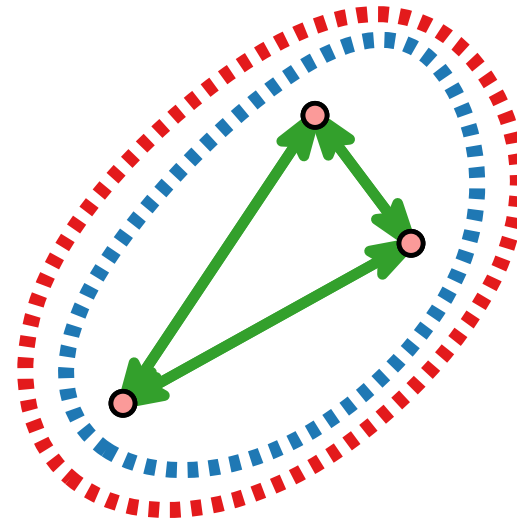
Kerne

direkt dichte-erreichbar

DBSCAN Clustering

DBSCAN* Clustering

DBSCAN*:
(und Kerne)



DBSCAN: Beispiel

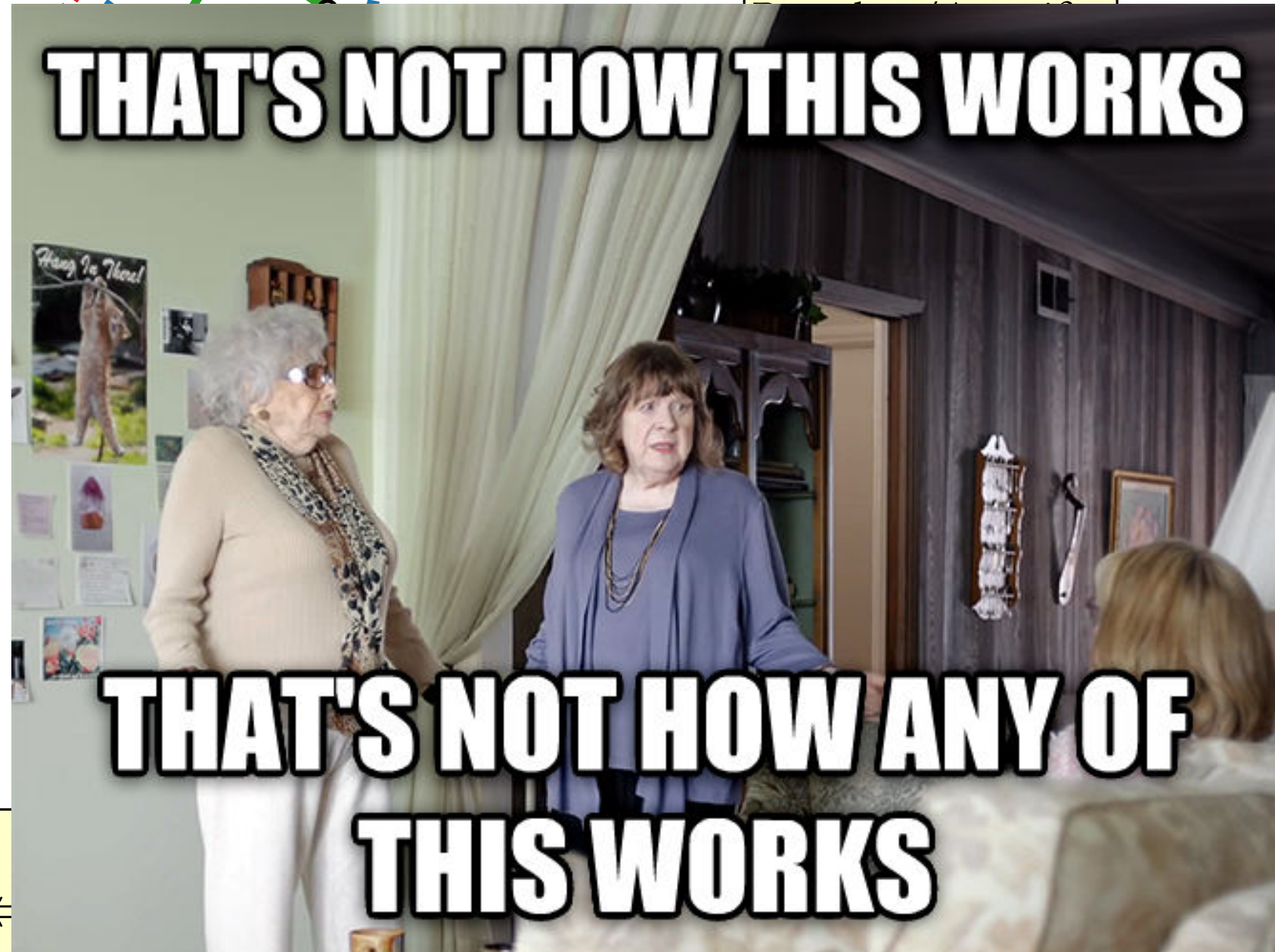
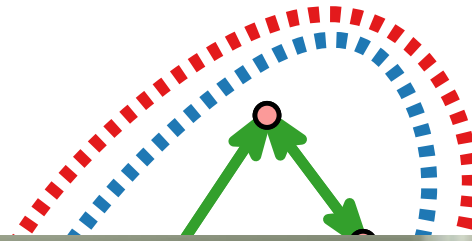
Laufzeit.

Naiver Algorithmus läuft in $\mathcal{O}(n^2)$ Zeit.

“Since the Eps-neighborhoods are expected to be small compared to the size of the whole data space, the average run time complexity of a single region query is $\mathcal{O}(\log n)$. [...] Thus, the average run time complexity of DBSCAN is $\mathcal{O}(n \log n)$.”

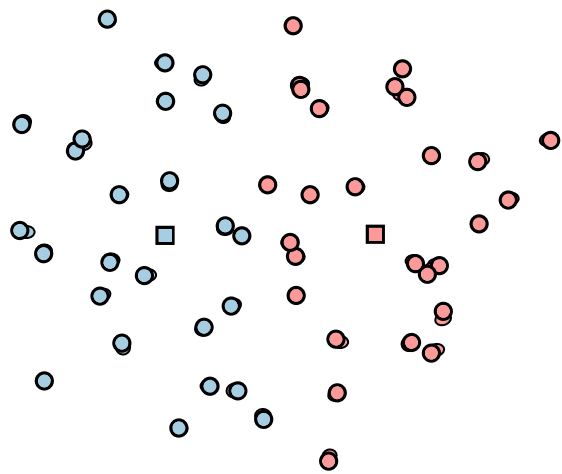
DBSCAN:

p und q sind im gleichen Cluster $\Leftrightarrow$

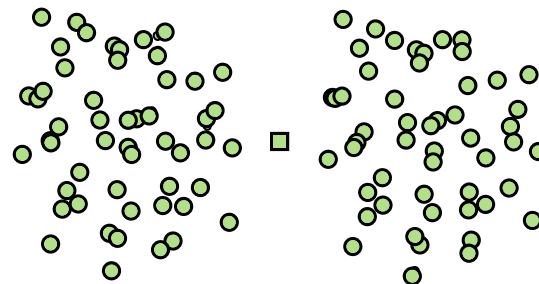
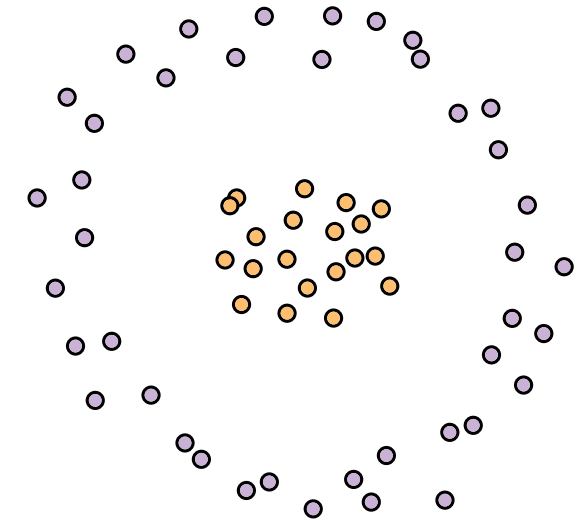


Algorithmen für geographische Informationssysteme

13. Vorlesung Clustering



Teil IV: DBSCAN: Laufzeit



De Berg, Gunawan, Roeloffzen (2017)



Annahme: ε frei, k feste Konstante, Euklidische Abstände

De Berg, Gunawan, Roeloffzen (2017)



Annahme: ε frei, k feste Konstante, Euklidische Abstände

2D

DBSCAN

$\mathcal{O}(n \log n)$

De Berg, Gunawan, Roeloffzen (2017)



Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D	d D	
DBSCAN	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$

De Berg, Gunawan, Roeloffzen (2017)



Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D	d D	
DBSCAN	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$
HDBSCAN	$\mathcal{O}(n \log n)$ expected	$\times$	

De Berg, Gunawan, Roeloffzen (2017)

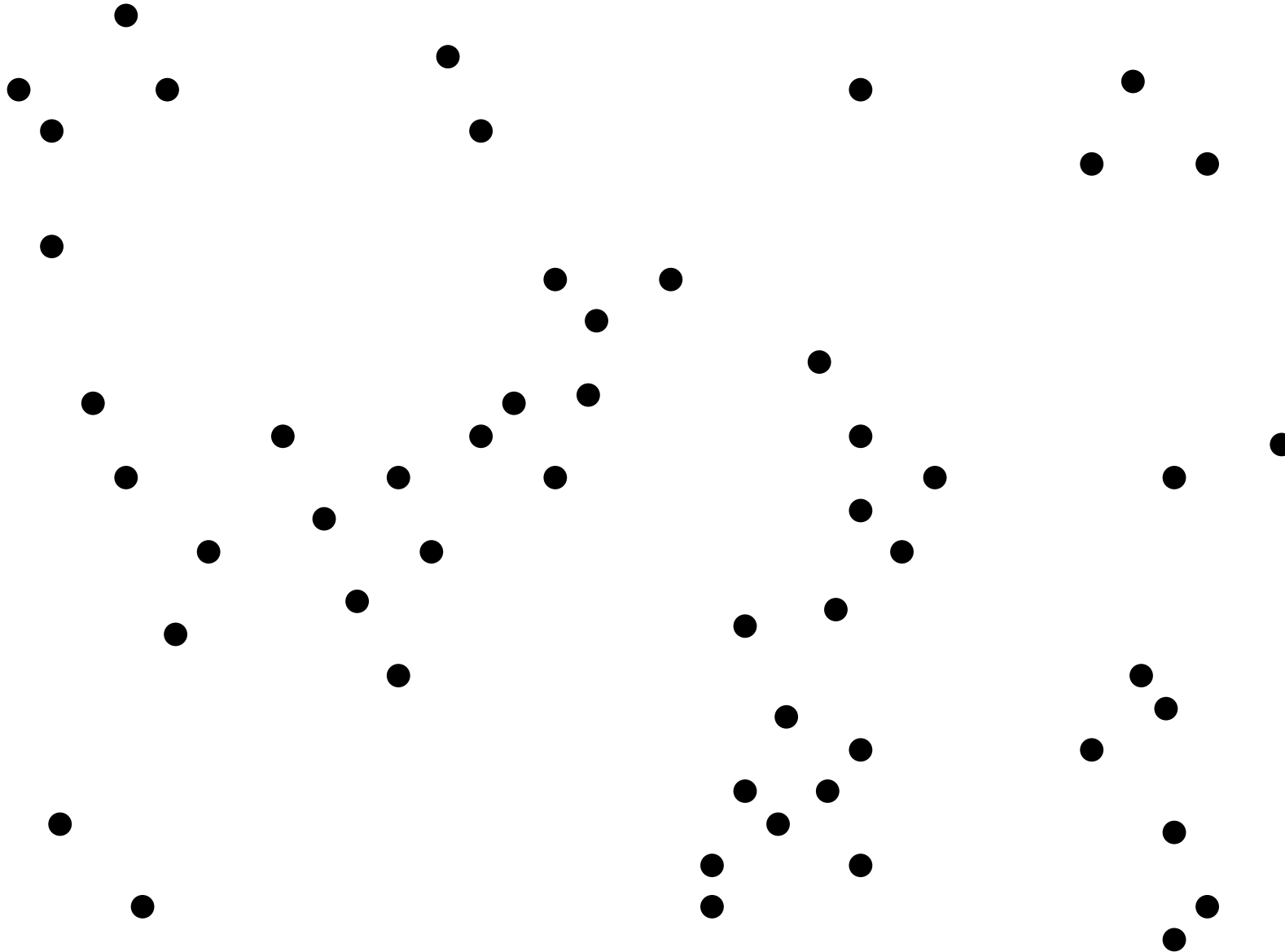


Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D		d D	
DBSCAN	$\mathcal{O}(n \log n)$		$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$
HDBSCAN	$\mathcal{O}(n \log n)$ expected		$\times$	

Box Graph $\mathcal{G}_{\text{box}}$

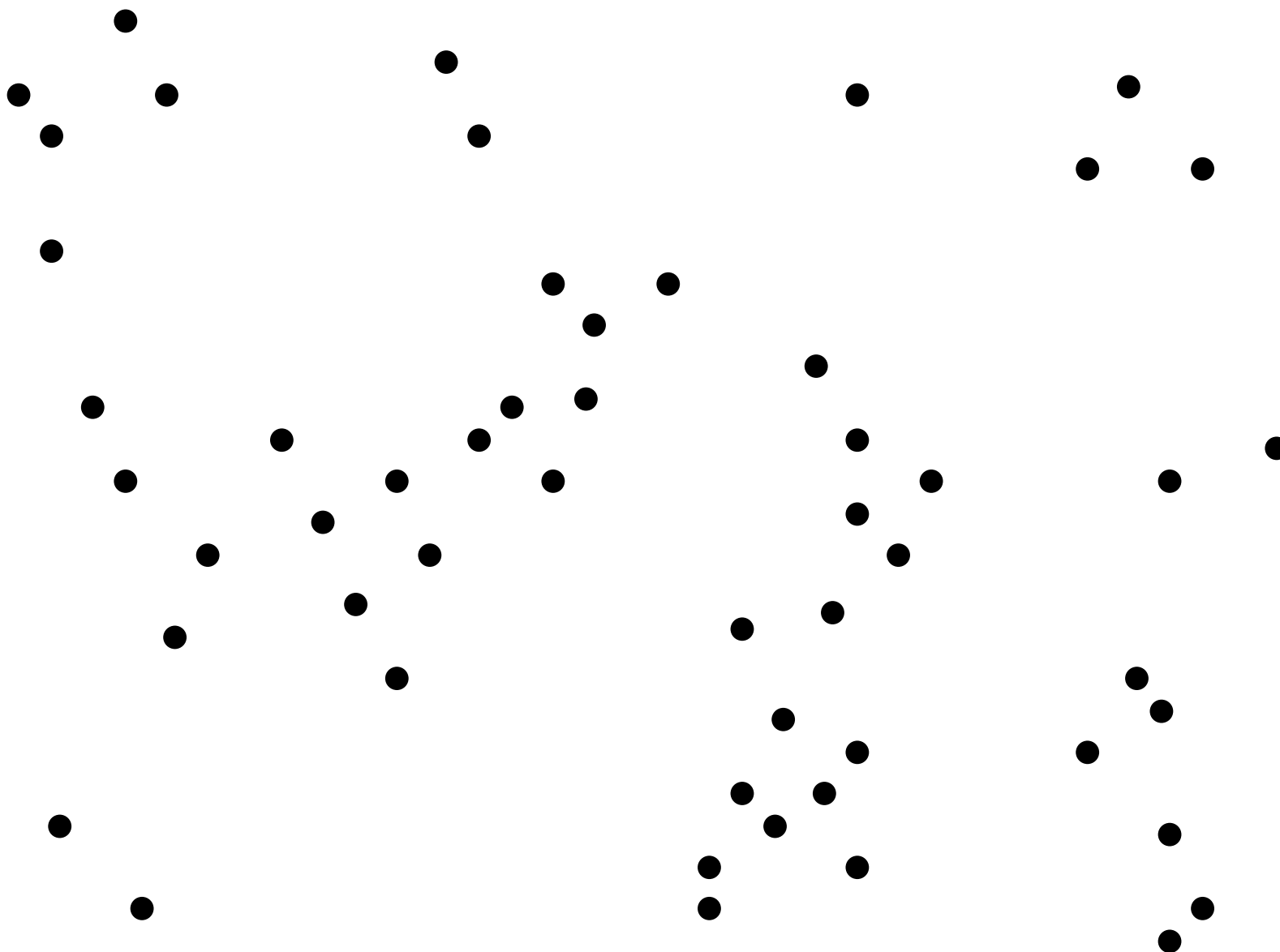
$k = 4$



Box Graph $\mathcal{G}_{\text{box}}$

$k = 4$

ε :



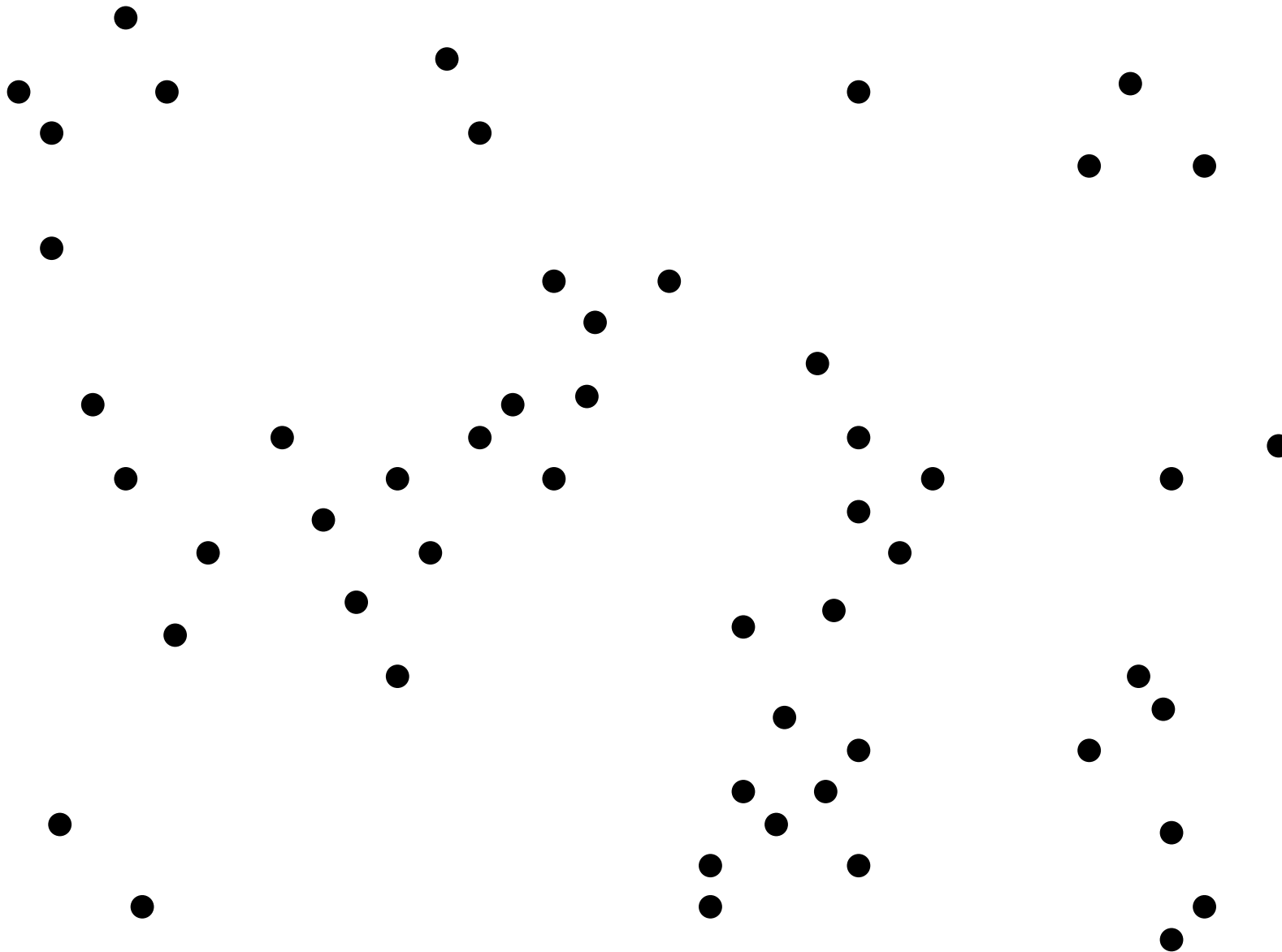
Box Graph $\mathcal{G}_{\text{box}}$

$k = 4$

ε : 



Gitterbasierter Ansatz.



Box Graph $\mathcal{G}_{\text{box}}$

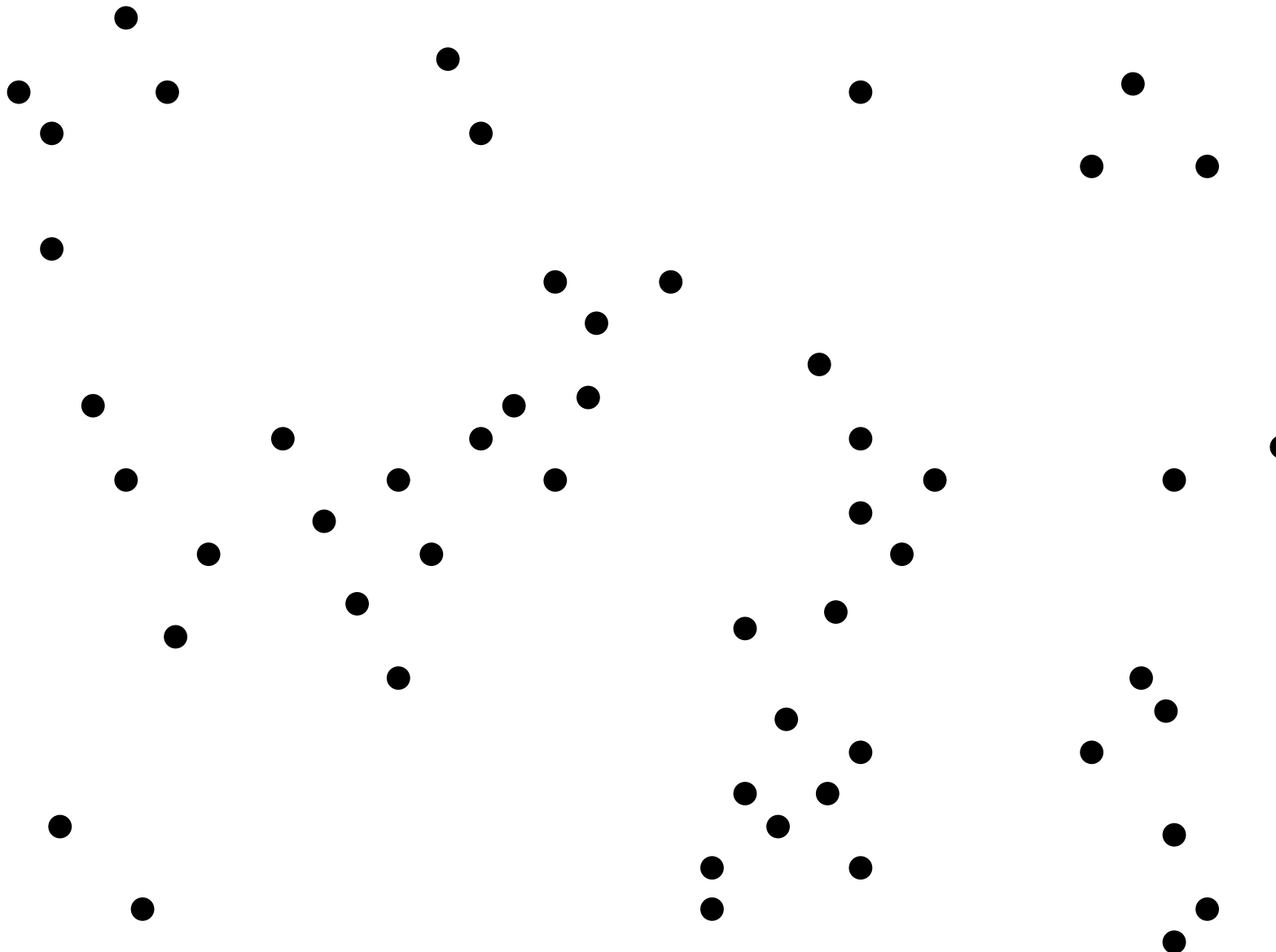
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Gitterbasierter Ansatz.

- Erstelle Gitter mit Zellengröße $\varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

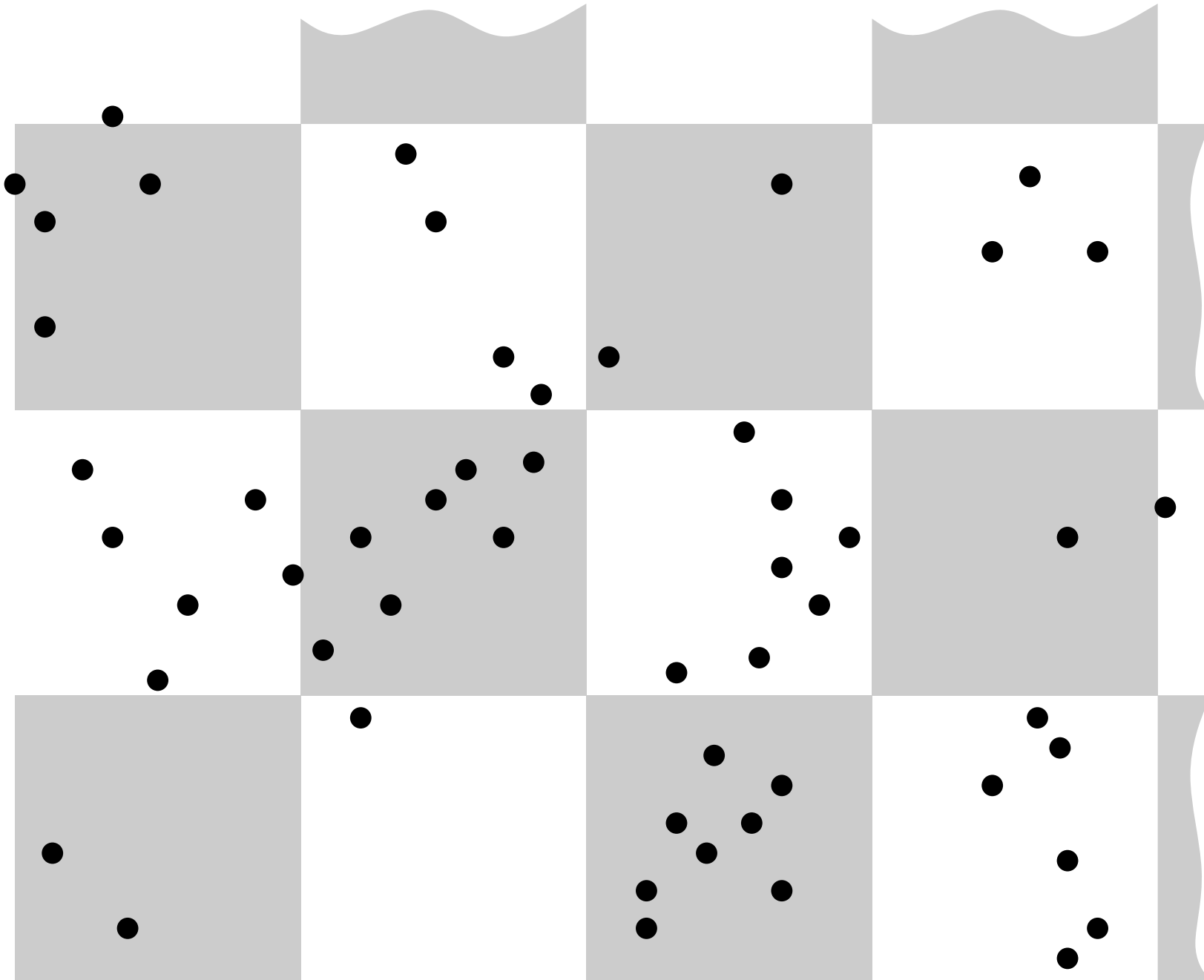
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Gitterbasierter Ansatz.

- Erstelle Gitter mit Zellengröße $\varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

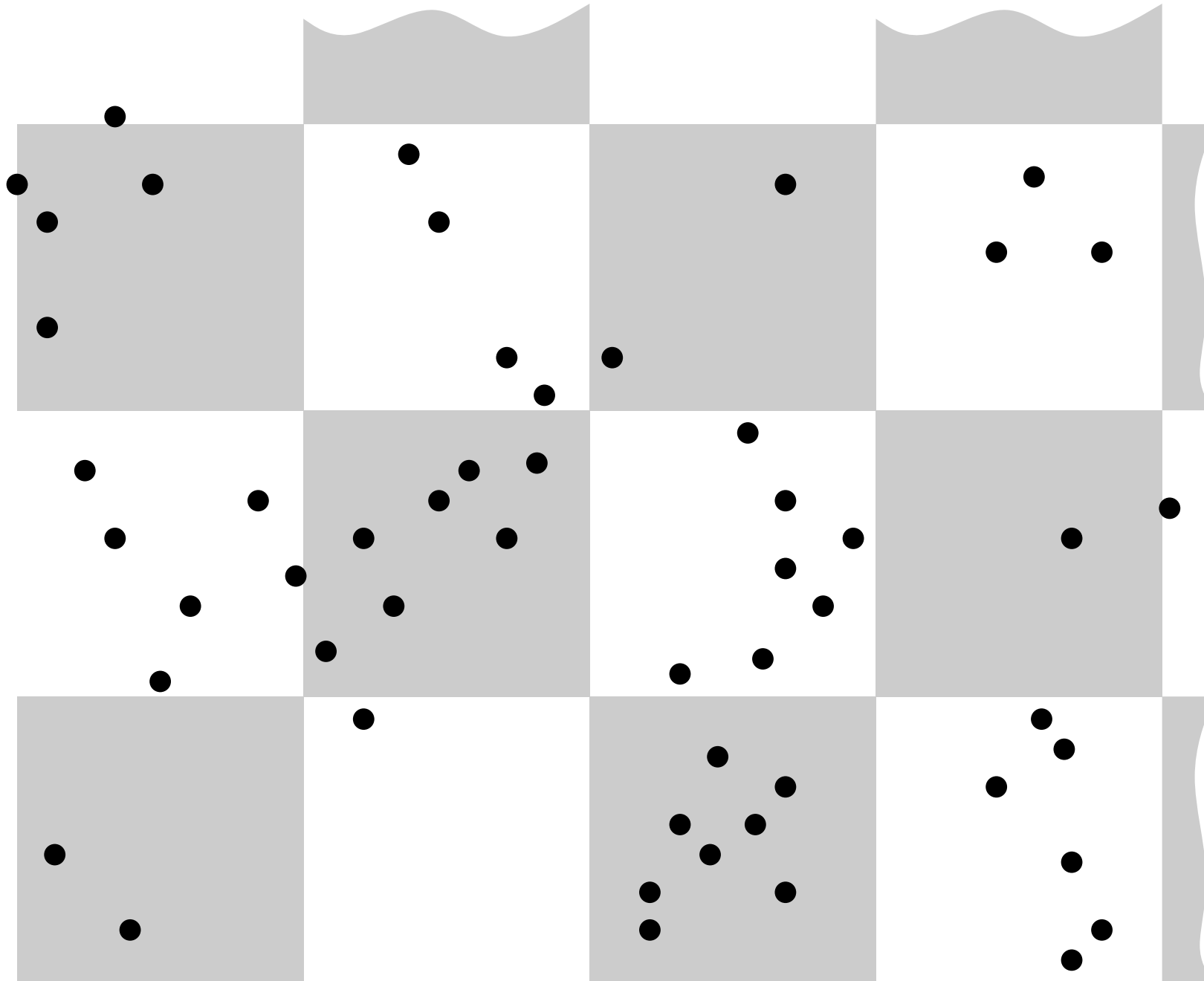
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Gitterbasierter Ansatz.

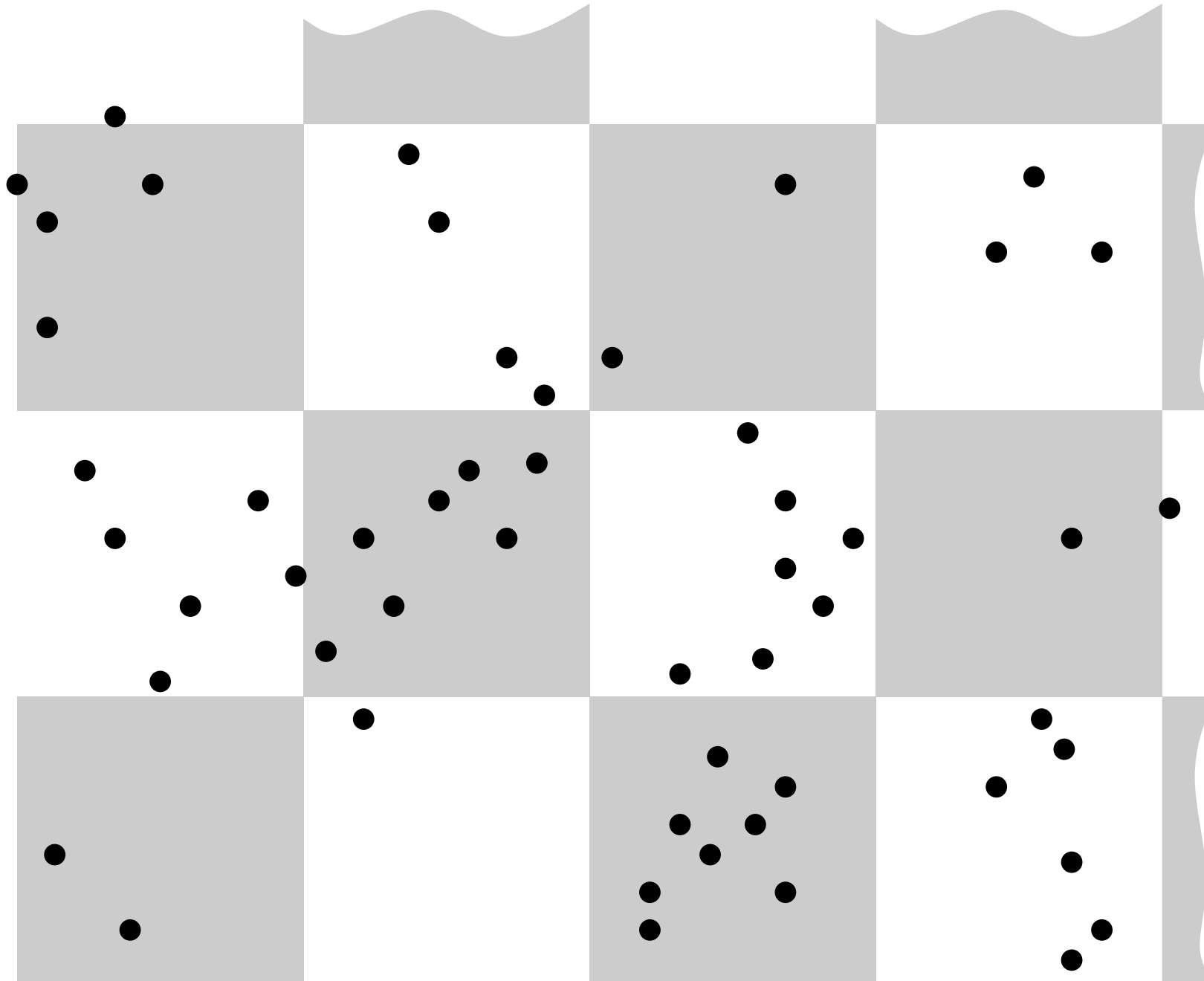
- Erstelle Gitter mit Zellengröße $\varepsilon/\sqrt{2}$.
- Zusammenhang in Gitterzellen?



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



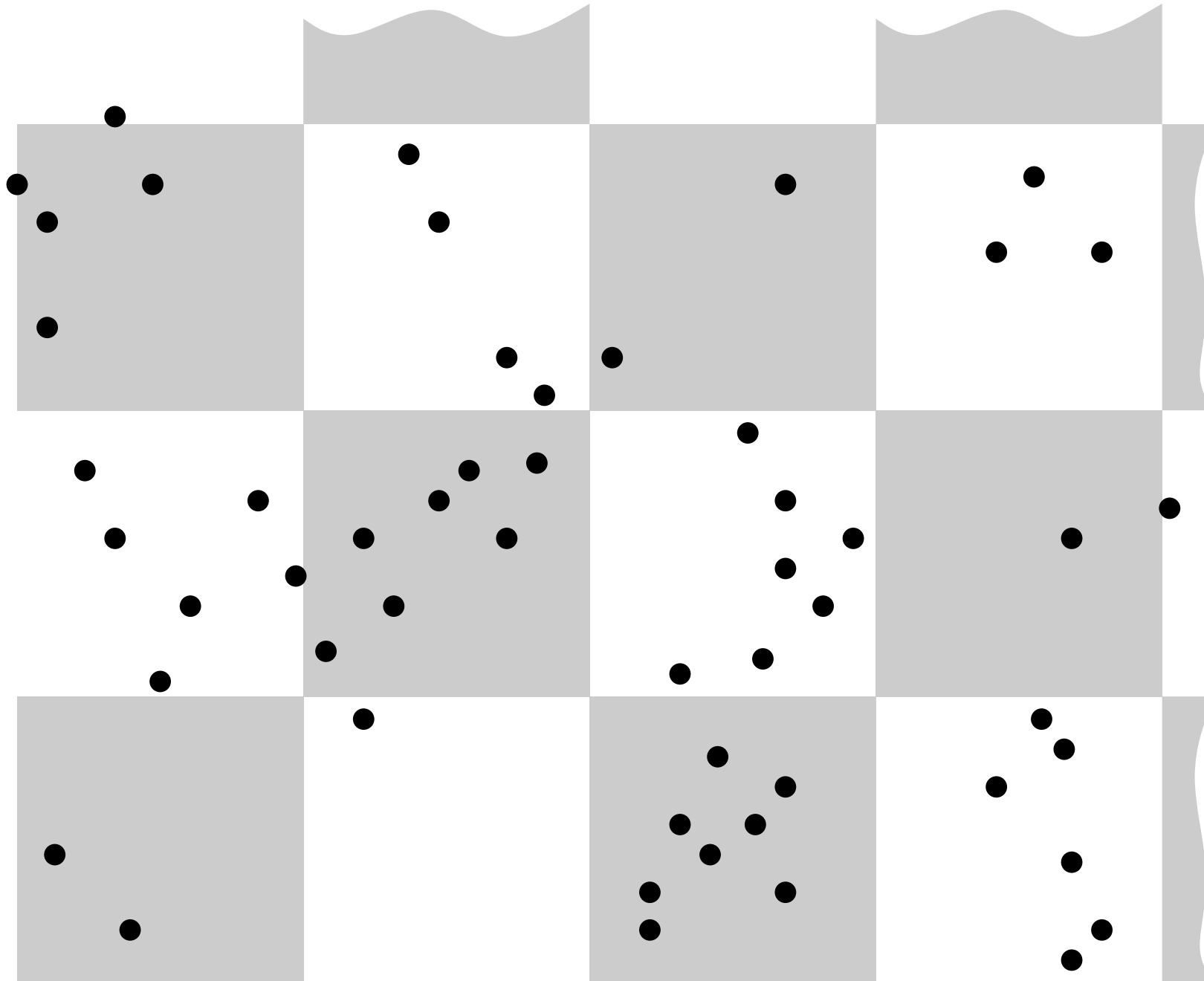
Gitterbasierter Ansatz.

- Erstelle Gitter mit Zellengröße $\varepsilon/\sqrt{2}$.
- Zusammenhang in Gitterzellen?
- ... zwischen Gitterzellen?

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



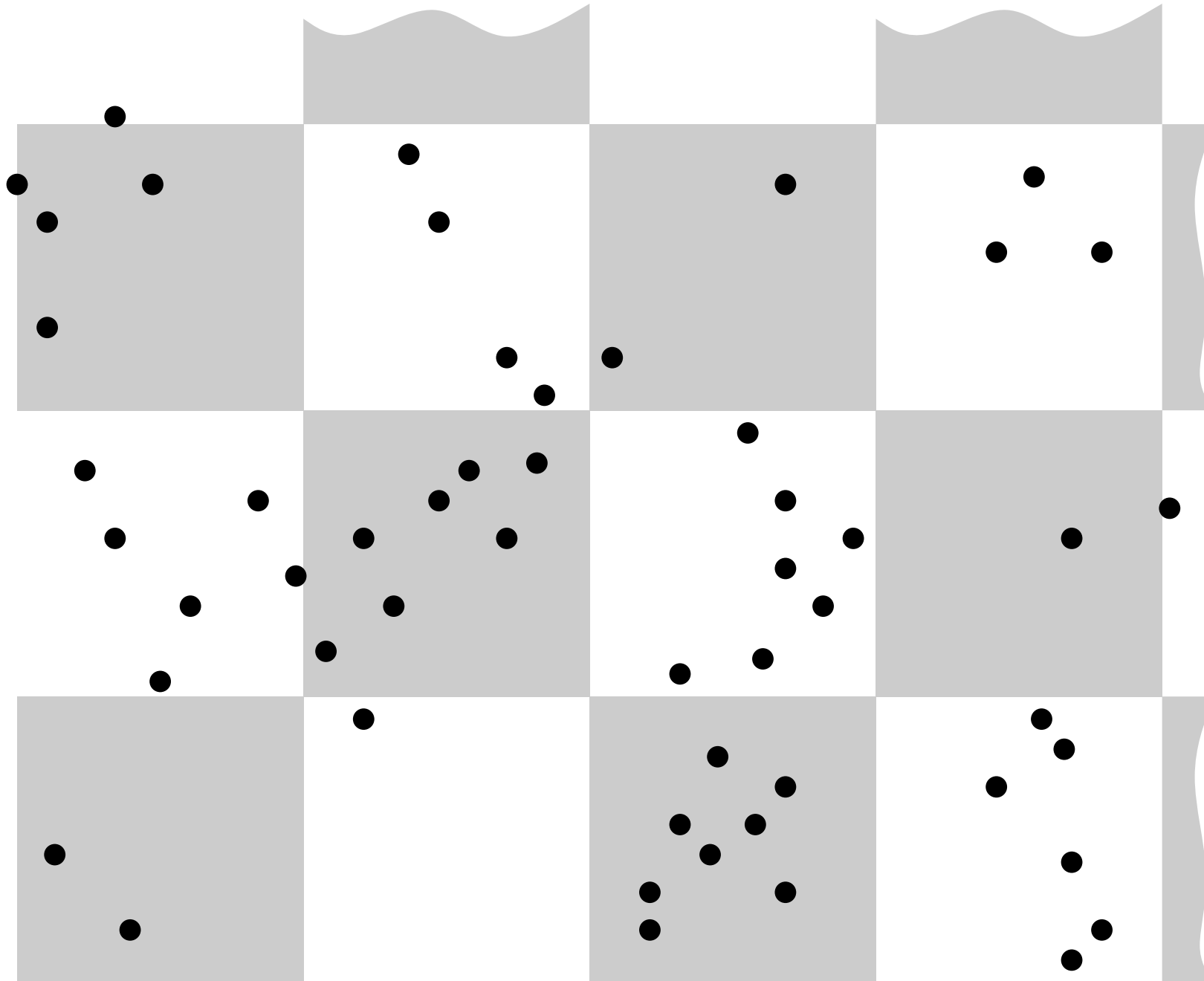
Gitterbasierter Ansatz.

- Erstelle Gitter mit Zellengröße $\varepsilon/\sqrt{2}$.
- Zusammenhang in Gitterzellen?
- ... zwischen Gitterzellen?
- Nicht klar, wie wir die Laufzeit abhängig von n berechnen können, ohne die Verteilung der Punkte zu kennen.

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Gitterbasierter Ansatz.

- Erstelle Gitter mit Zellengröße $\varepsilon/\sqrt{2}$.
- Zusammenhang in Gitterzellen?
- ... zwischen Gitterzellen?
- Nicht klar, wie wir die Laufzeit abhängig von n berechnen können, ohne die Verteilung der Punkte zu kennen.
- sei flexibler...

Box Graph $\mathcal{G}_{\text{box}}$

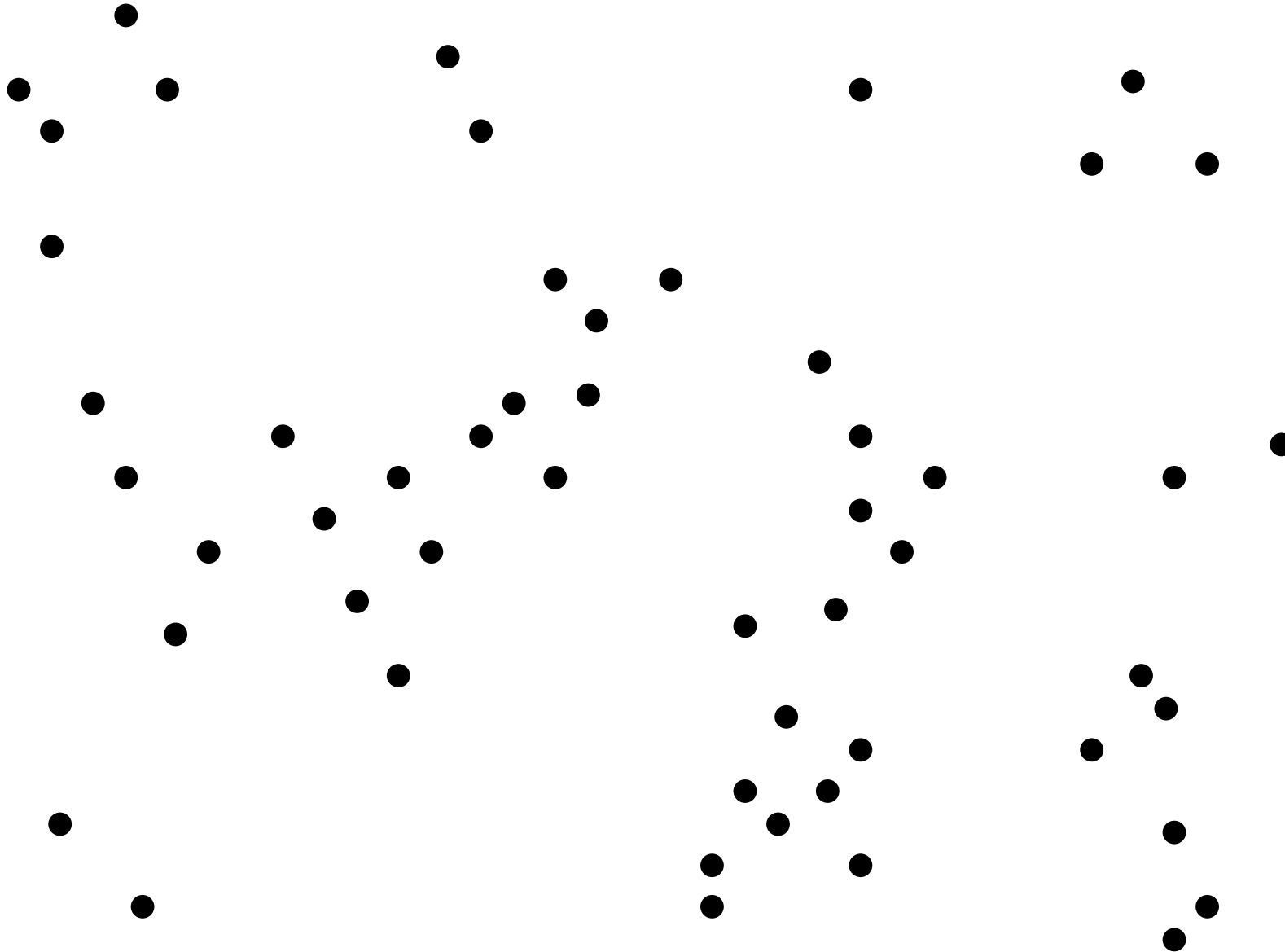
$$k = 4$$

$$\varepsilon:$$

 $\varepsilon/\sqrt{2}:$ 



1. Erstelle Boxen.



Box Graph $\mathcal{G}_{\text{box}}$

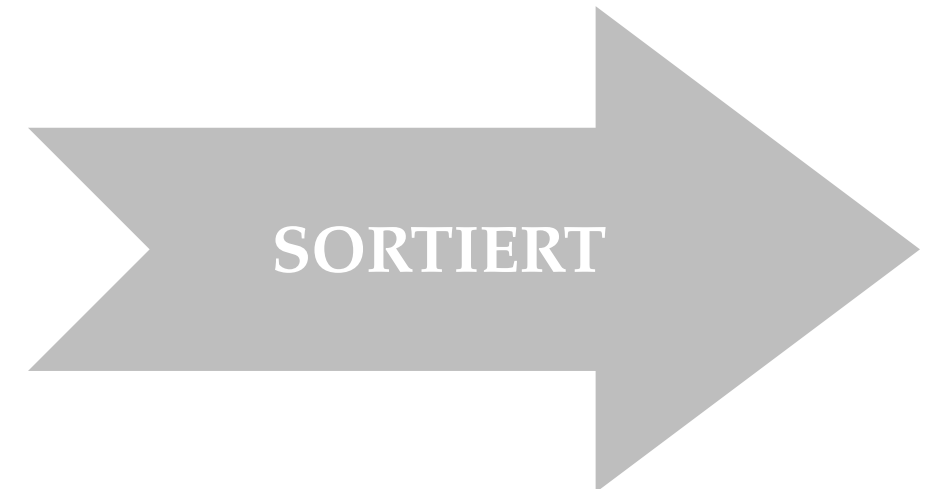
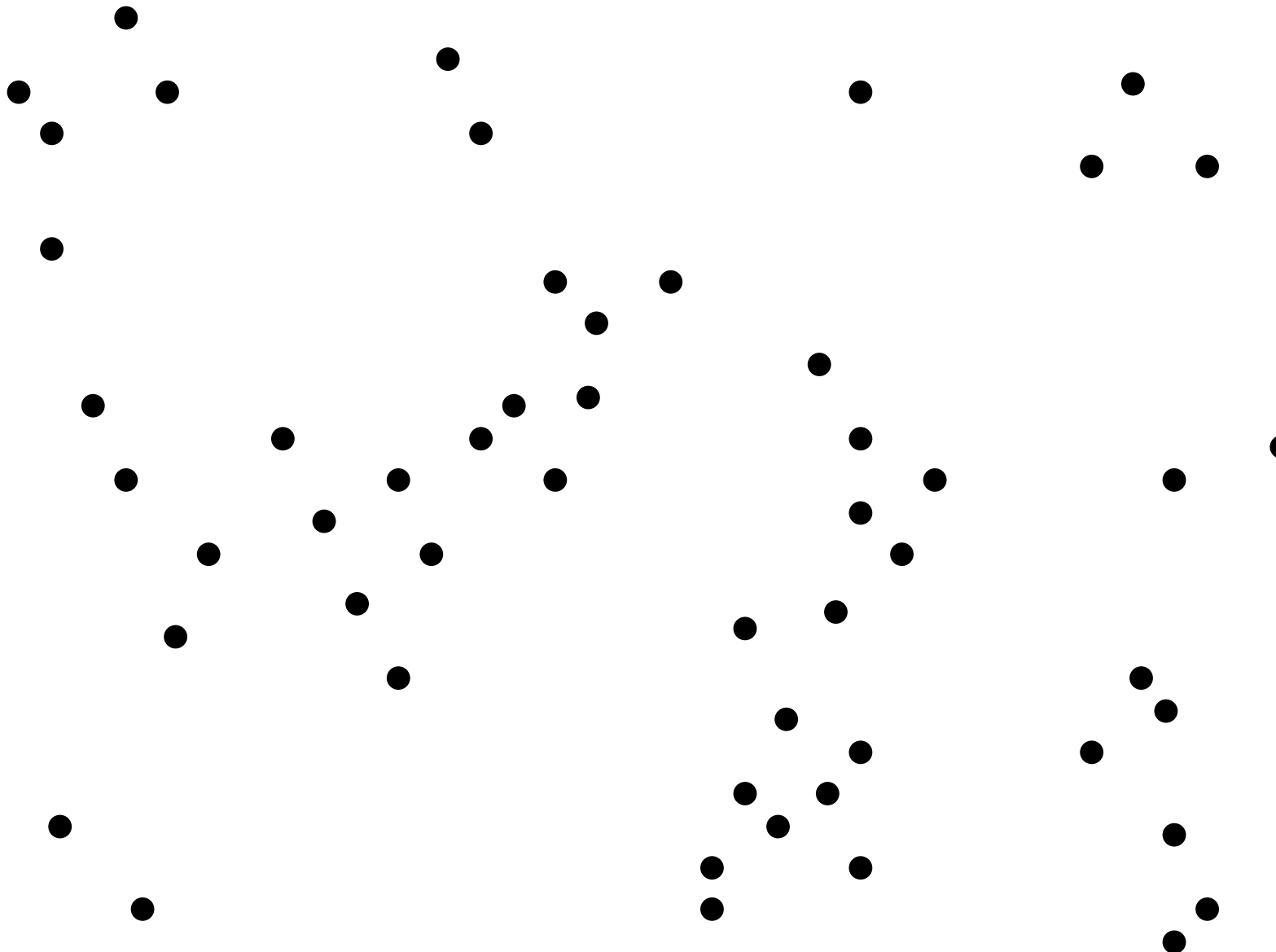
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

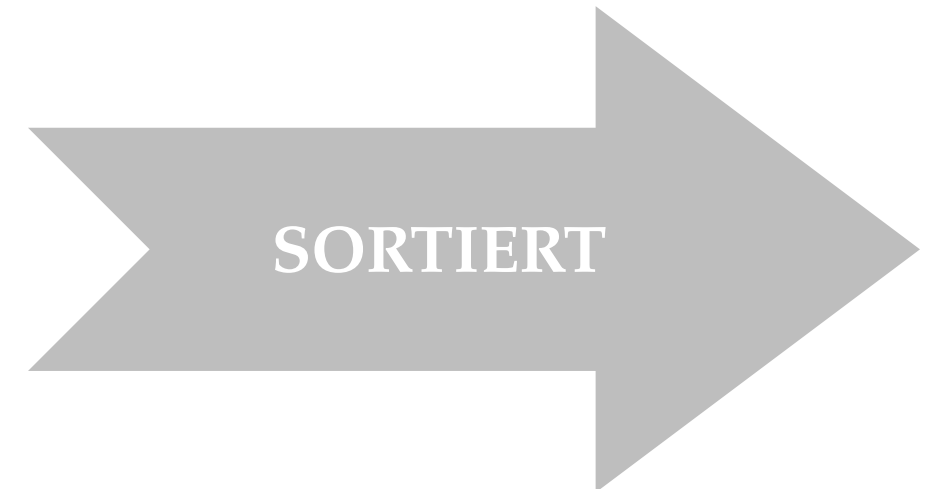
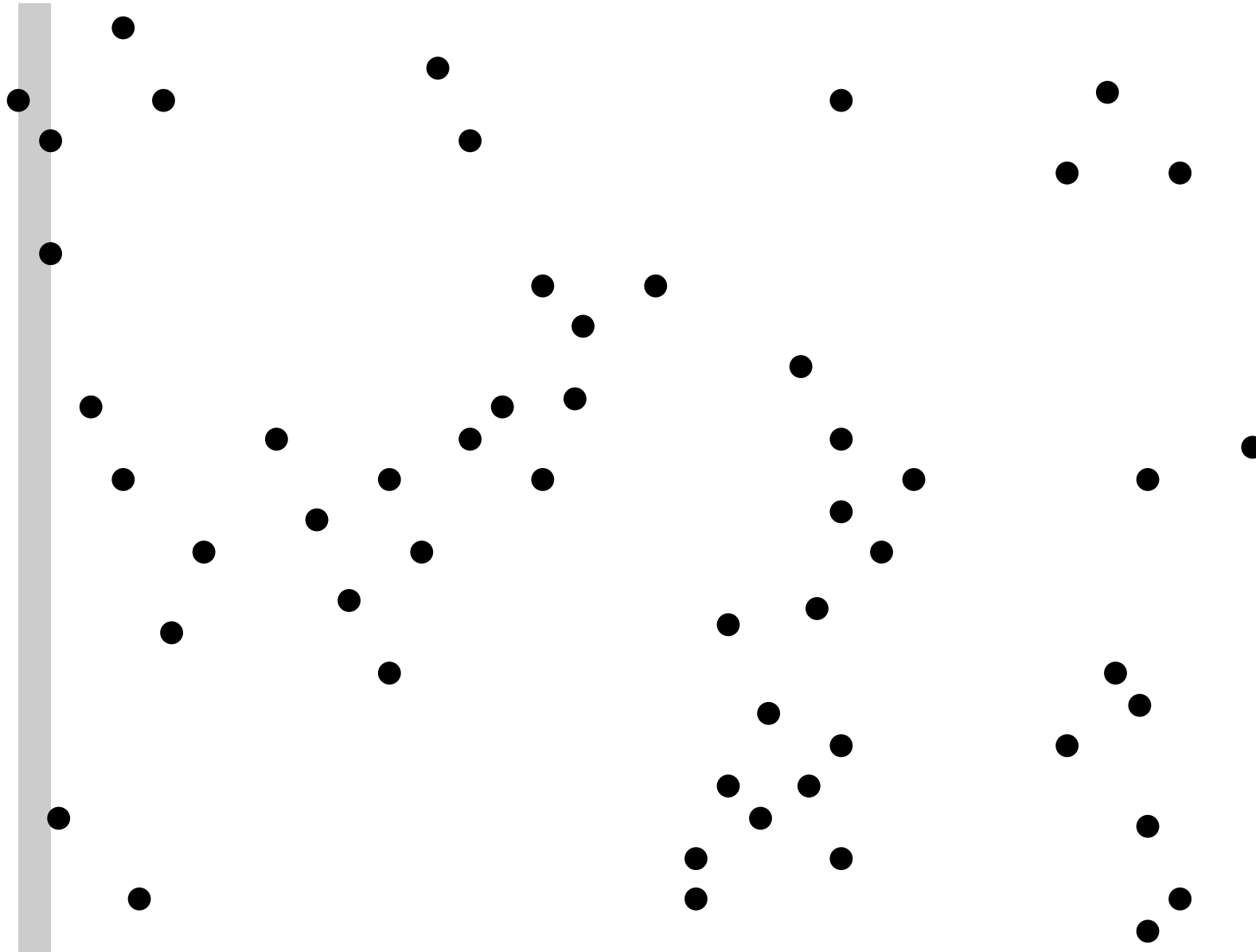
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

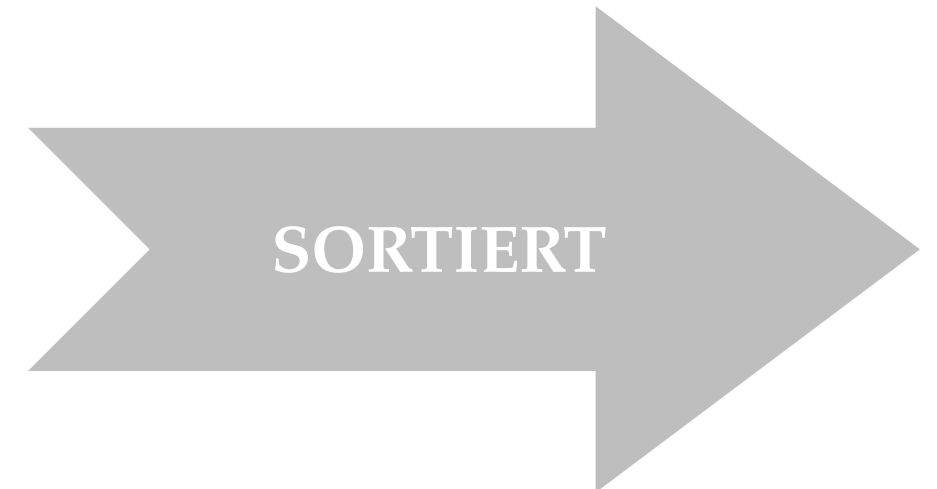
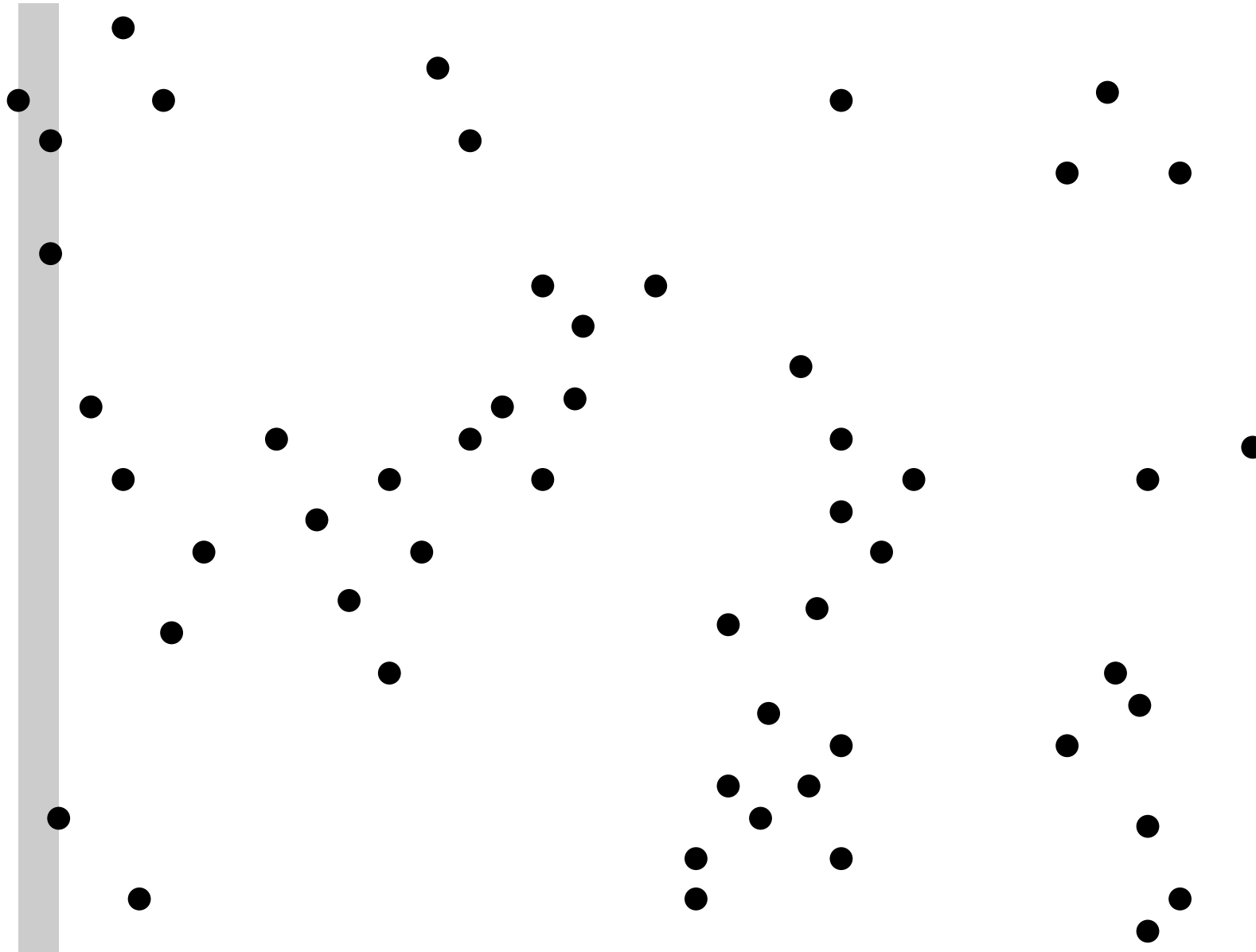
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

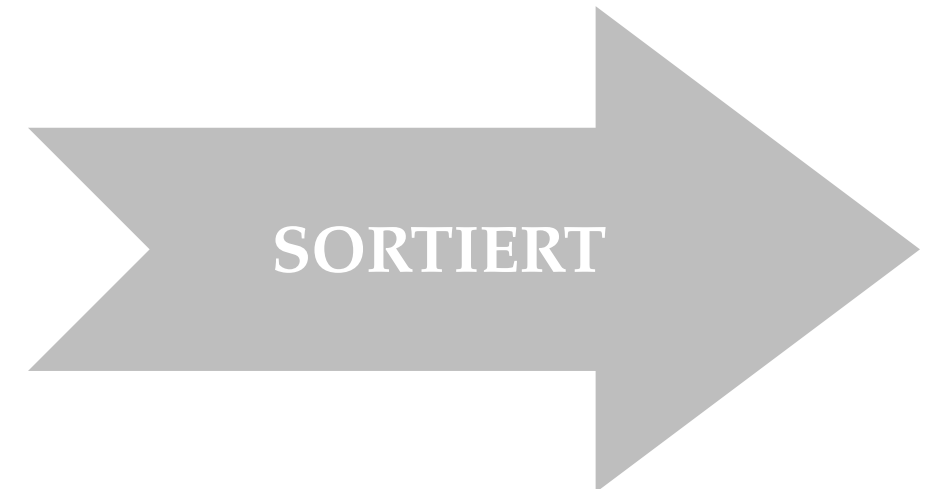
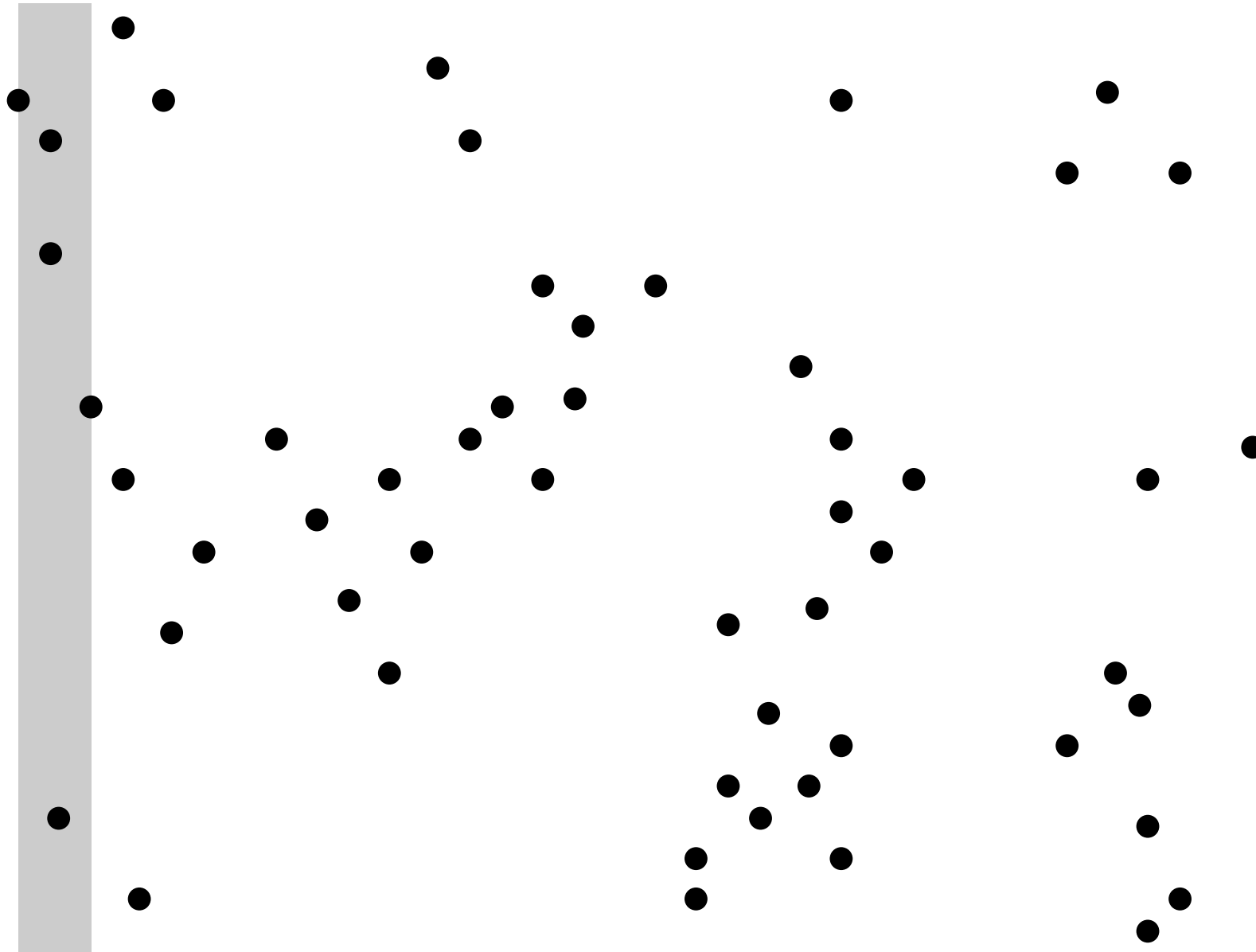
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

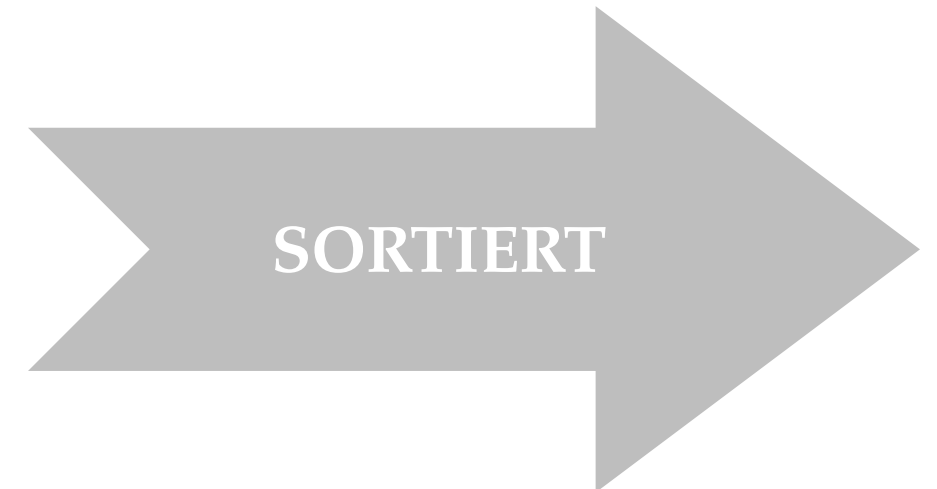
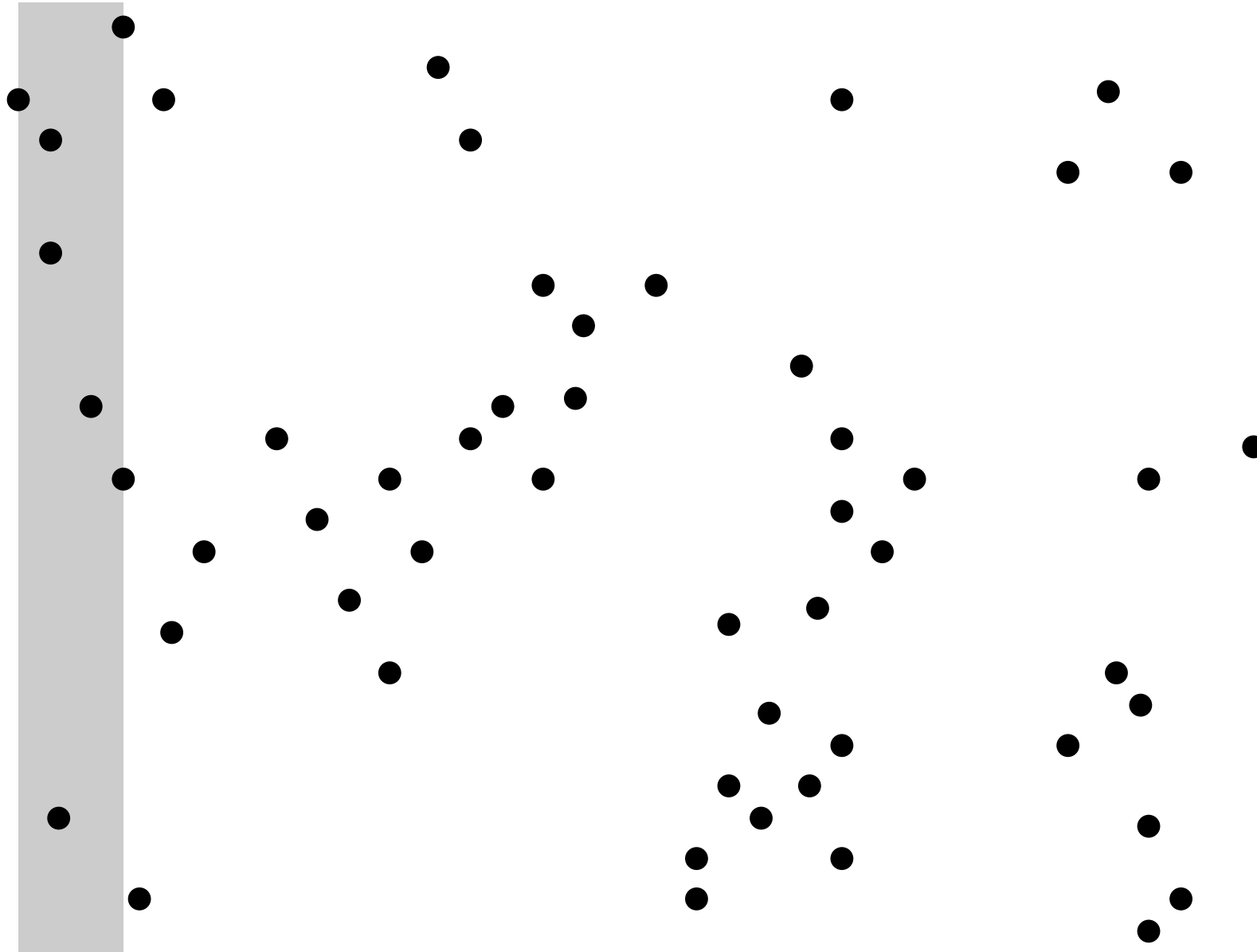
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

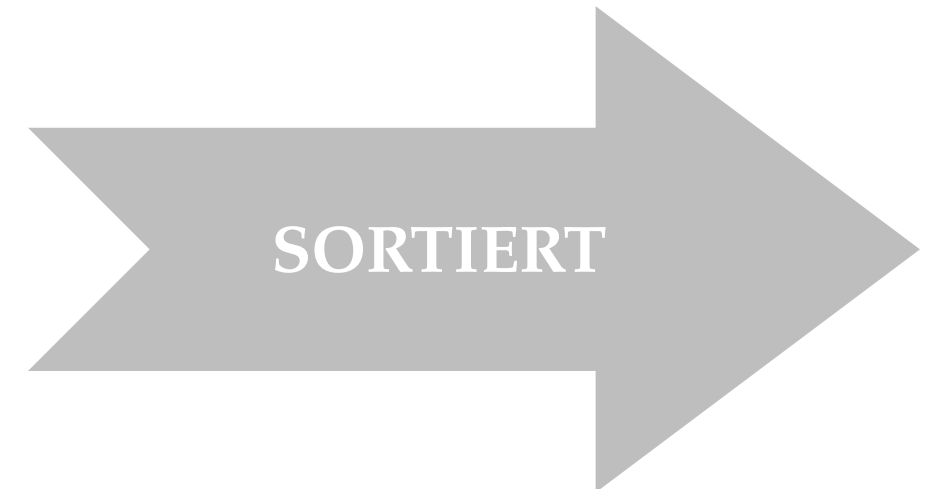
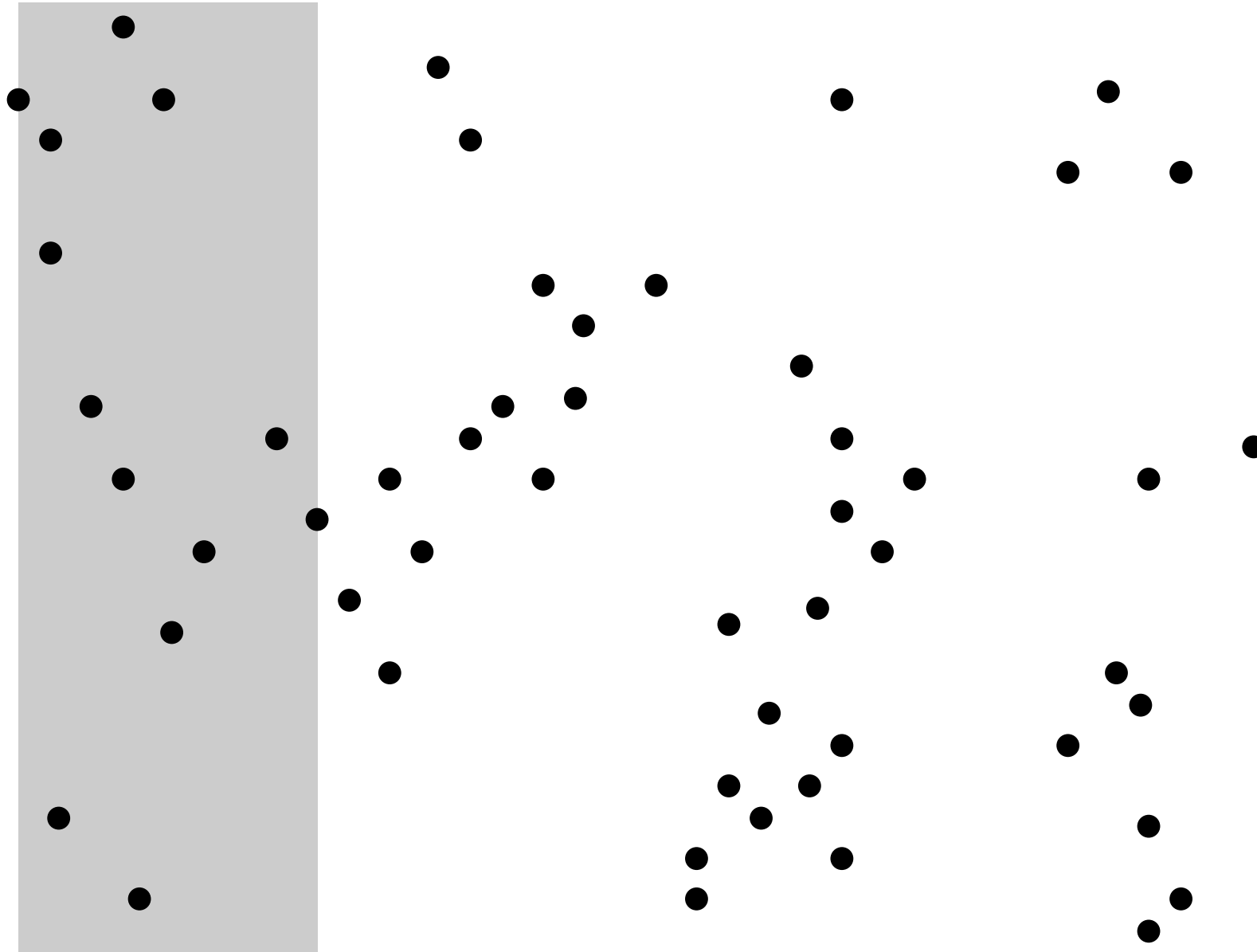
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

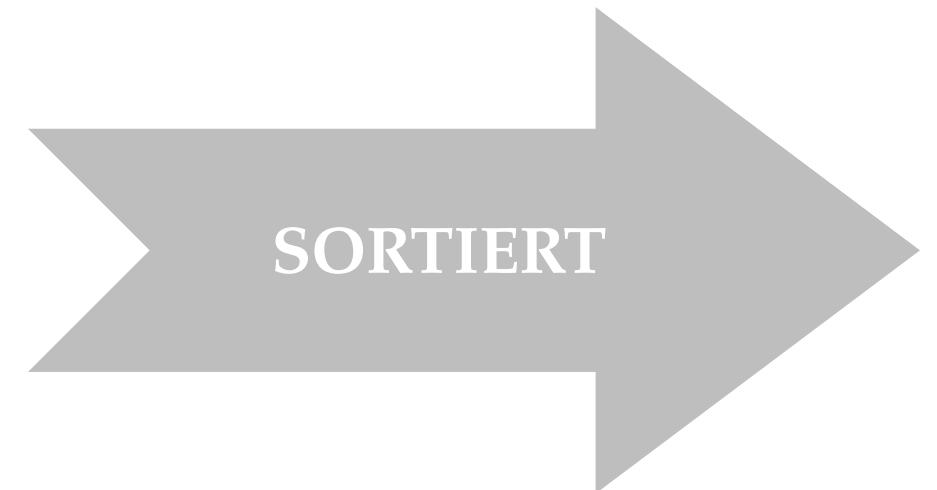
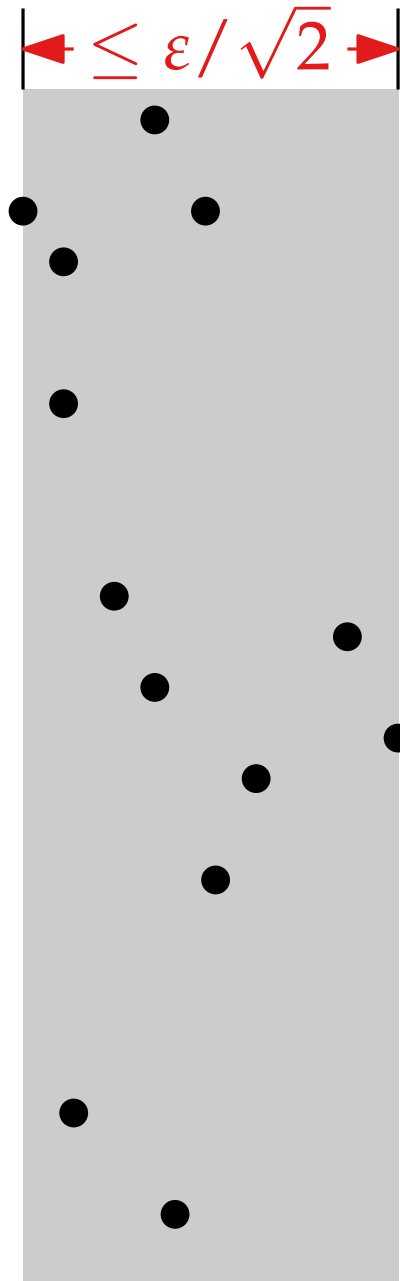
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

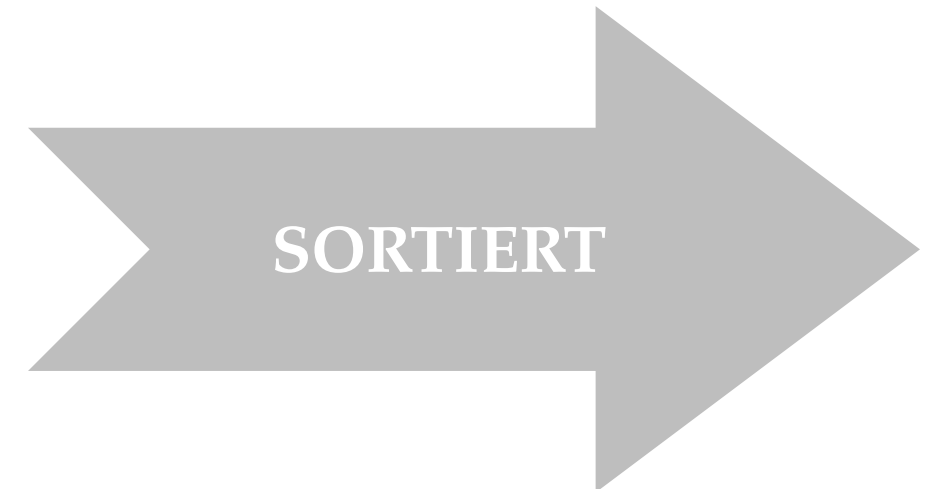
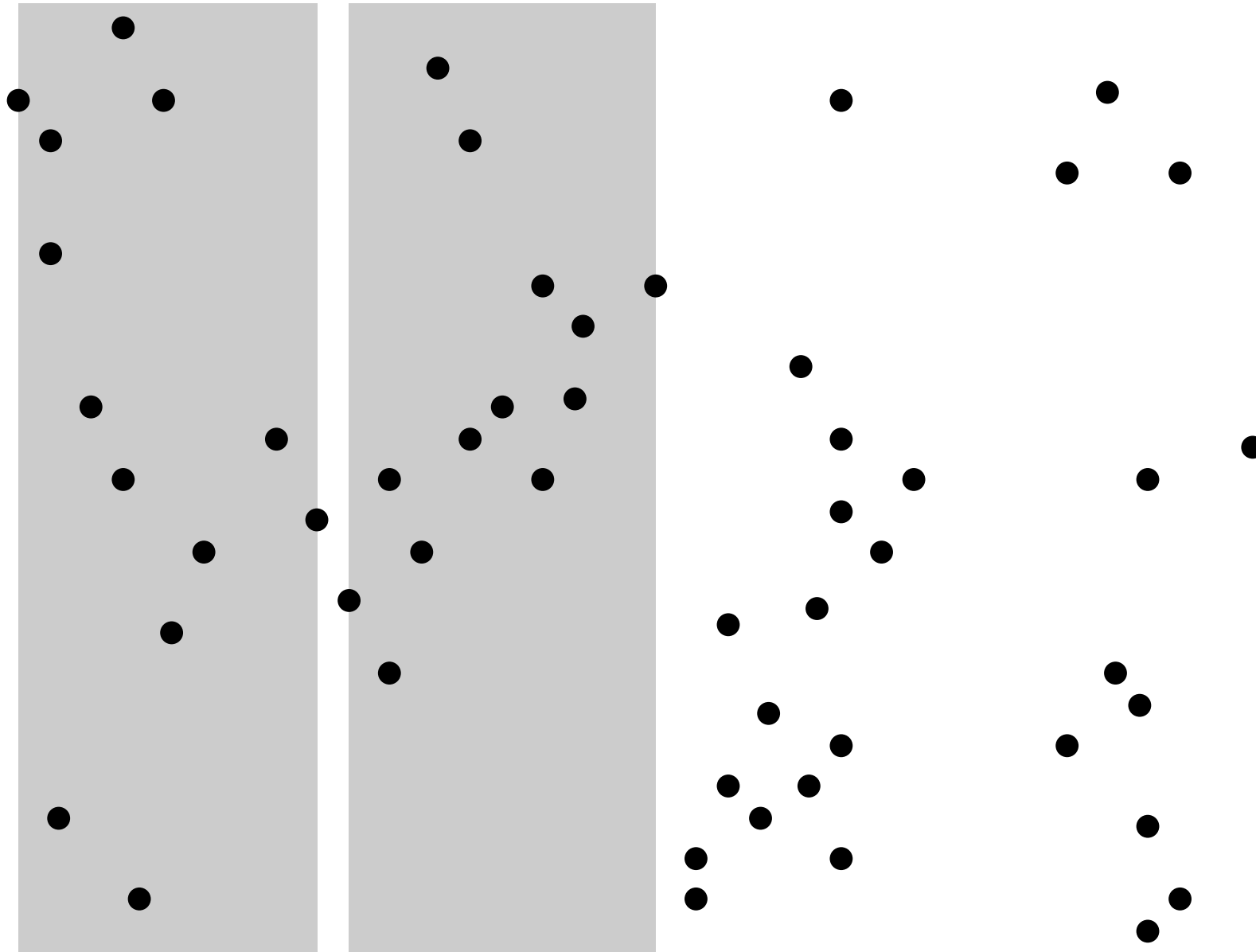
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

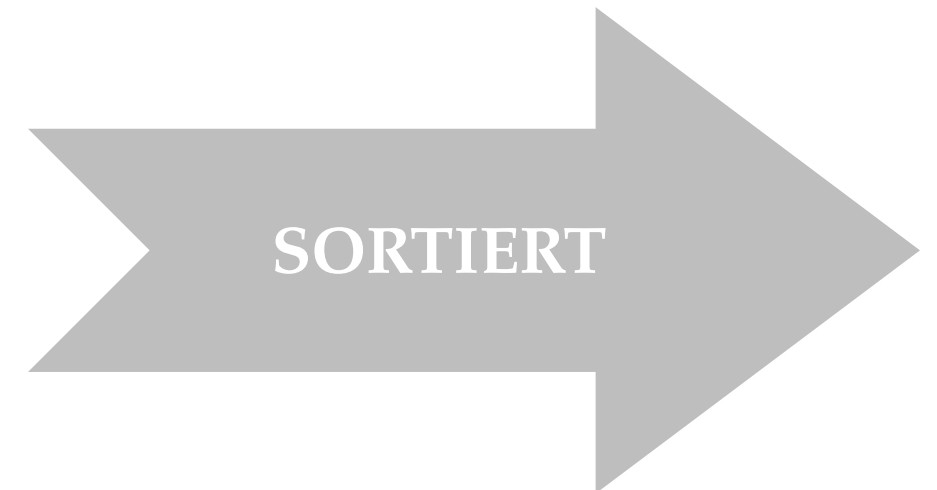
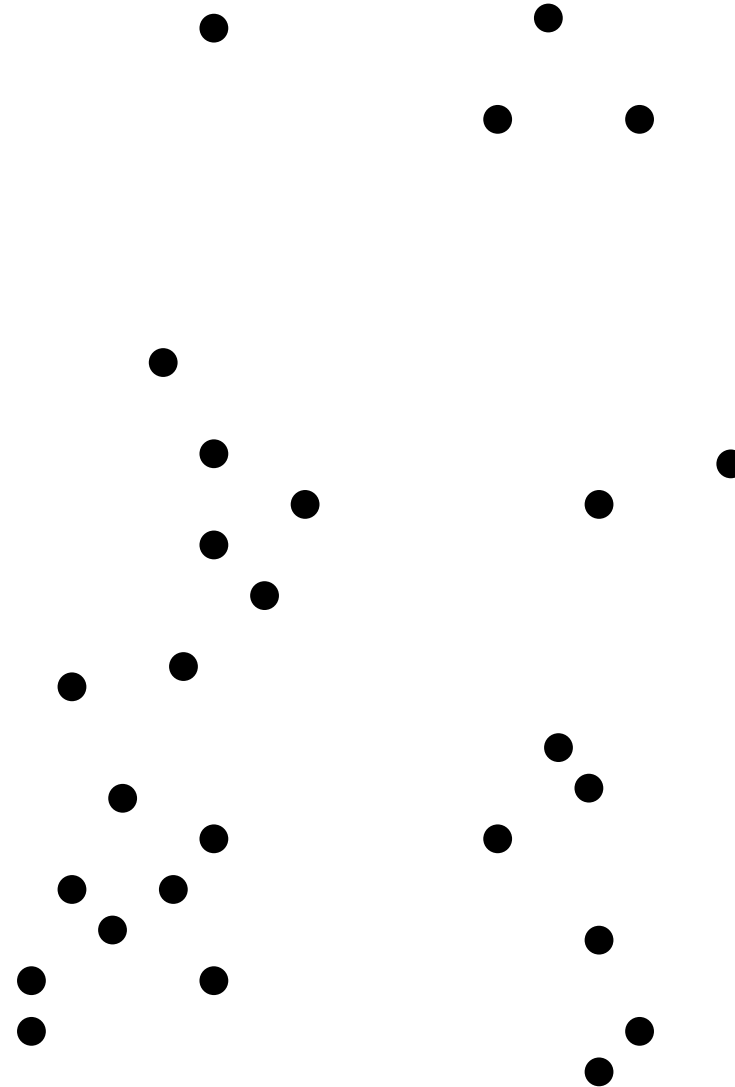
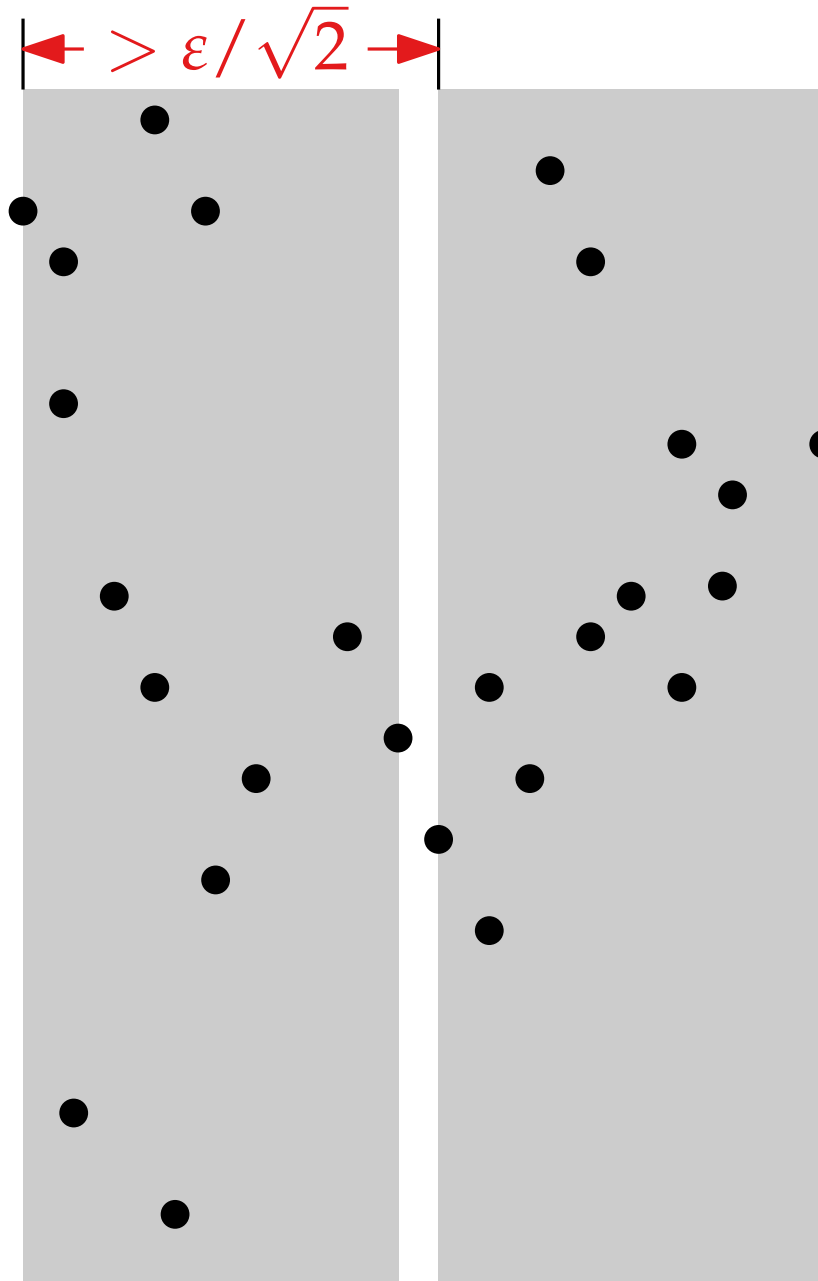
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

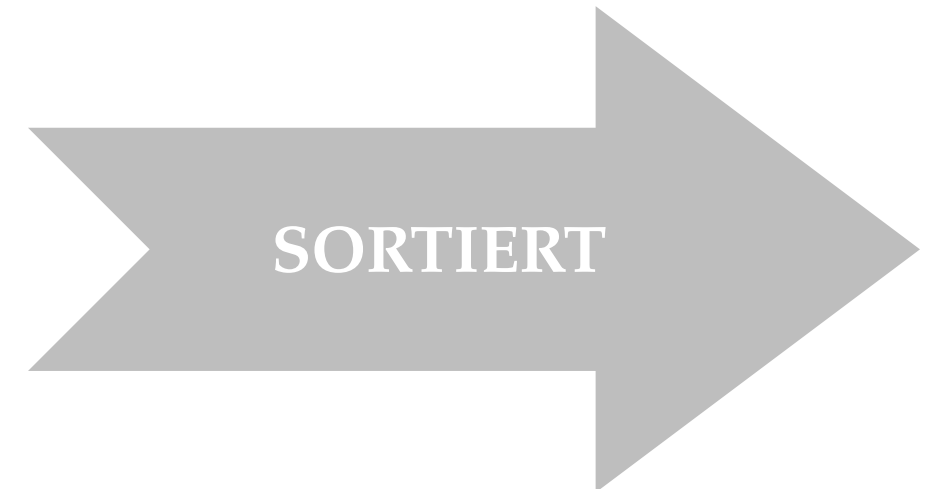
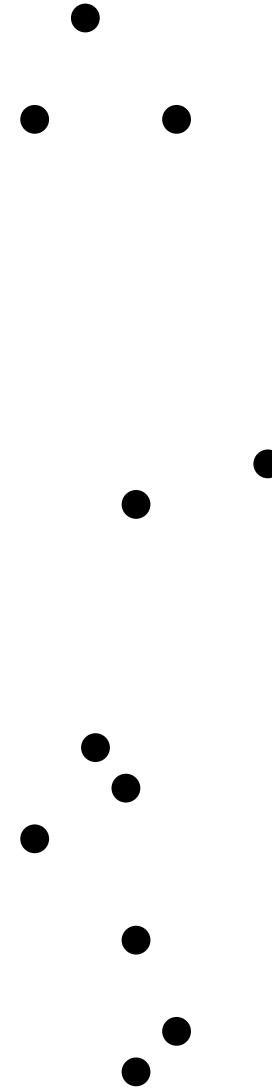
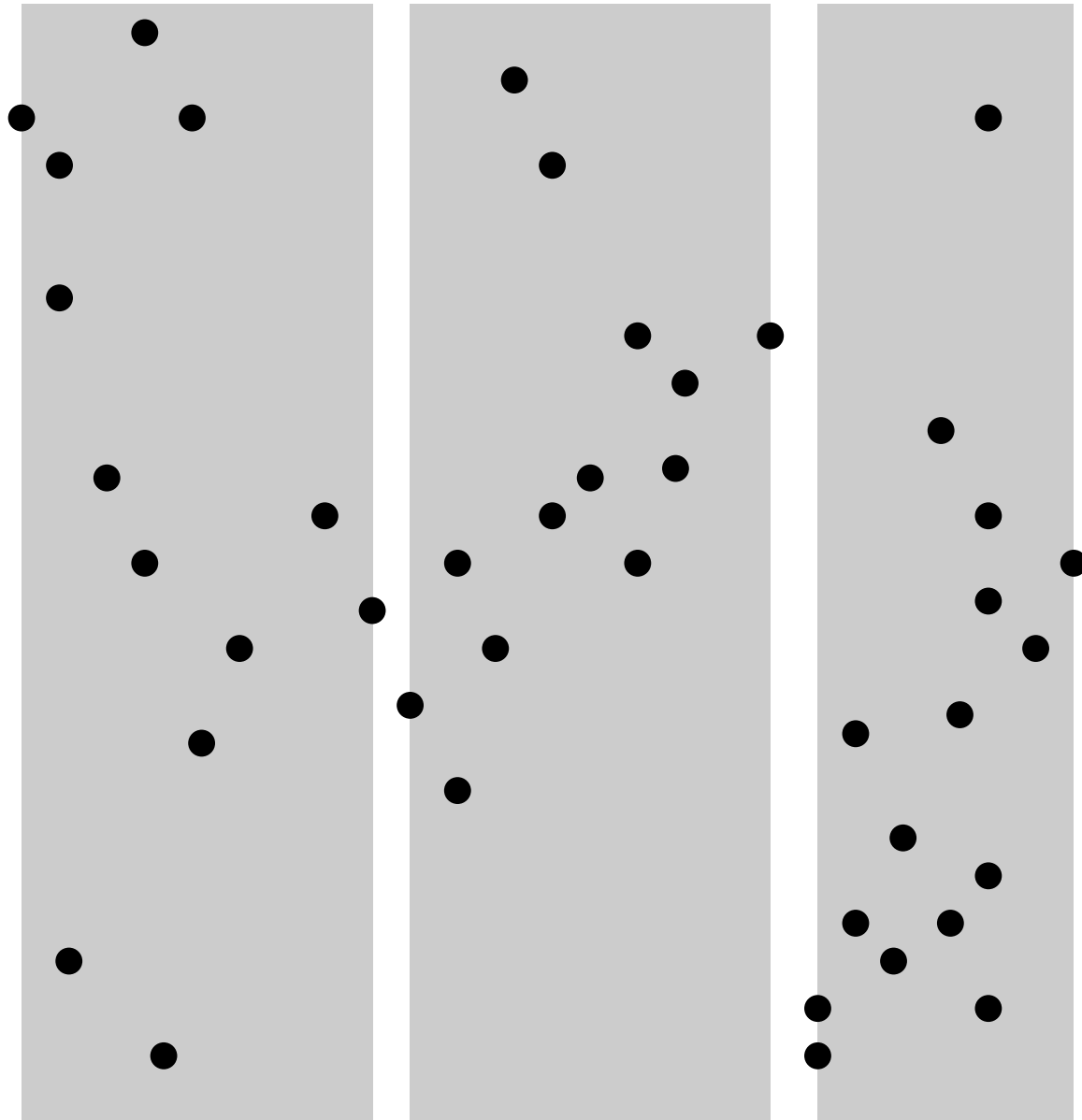
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

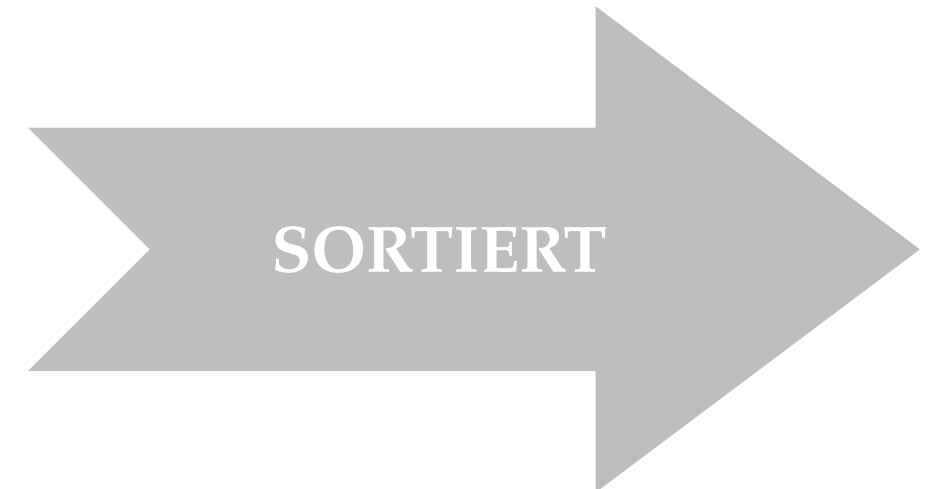
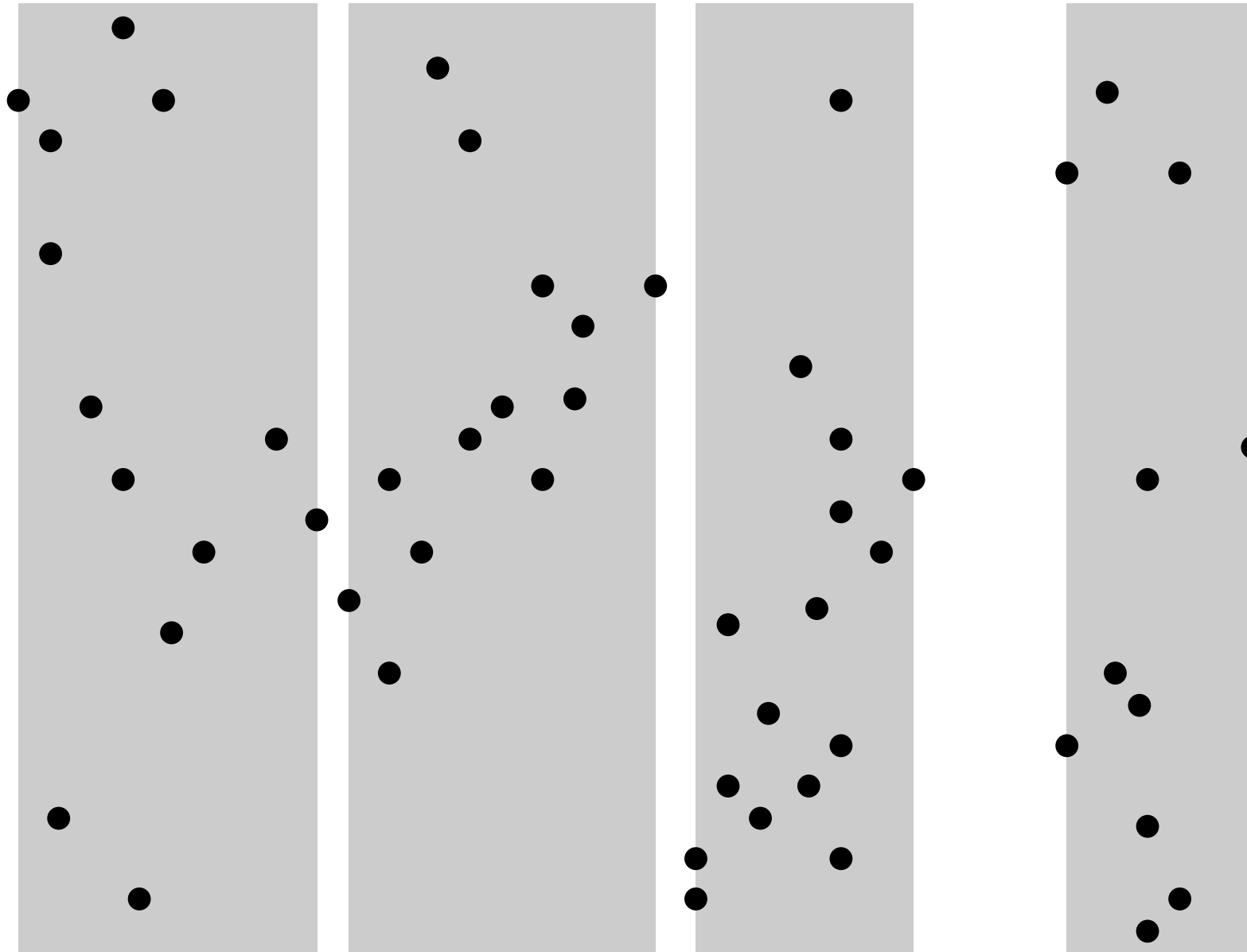
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

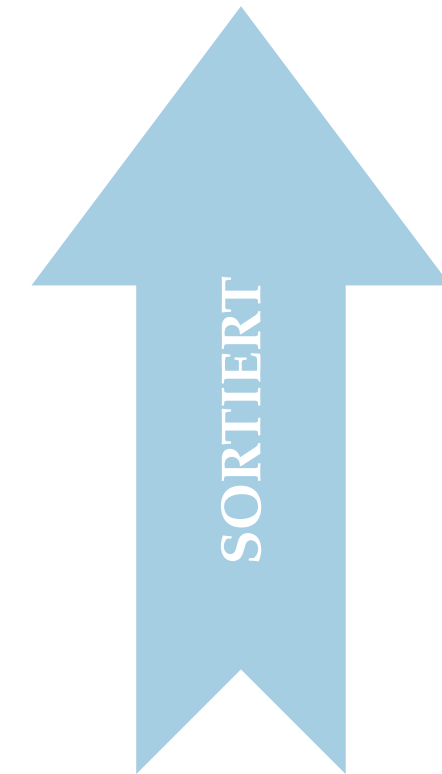
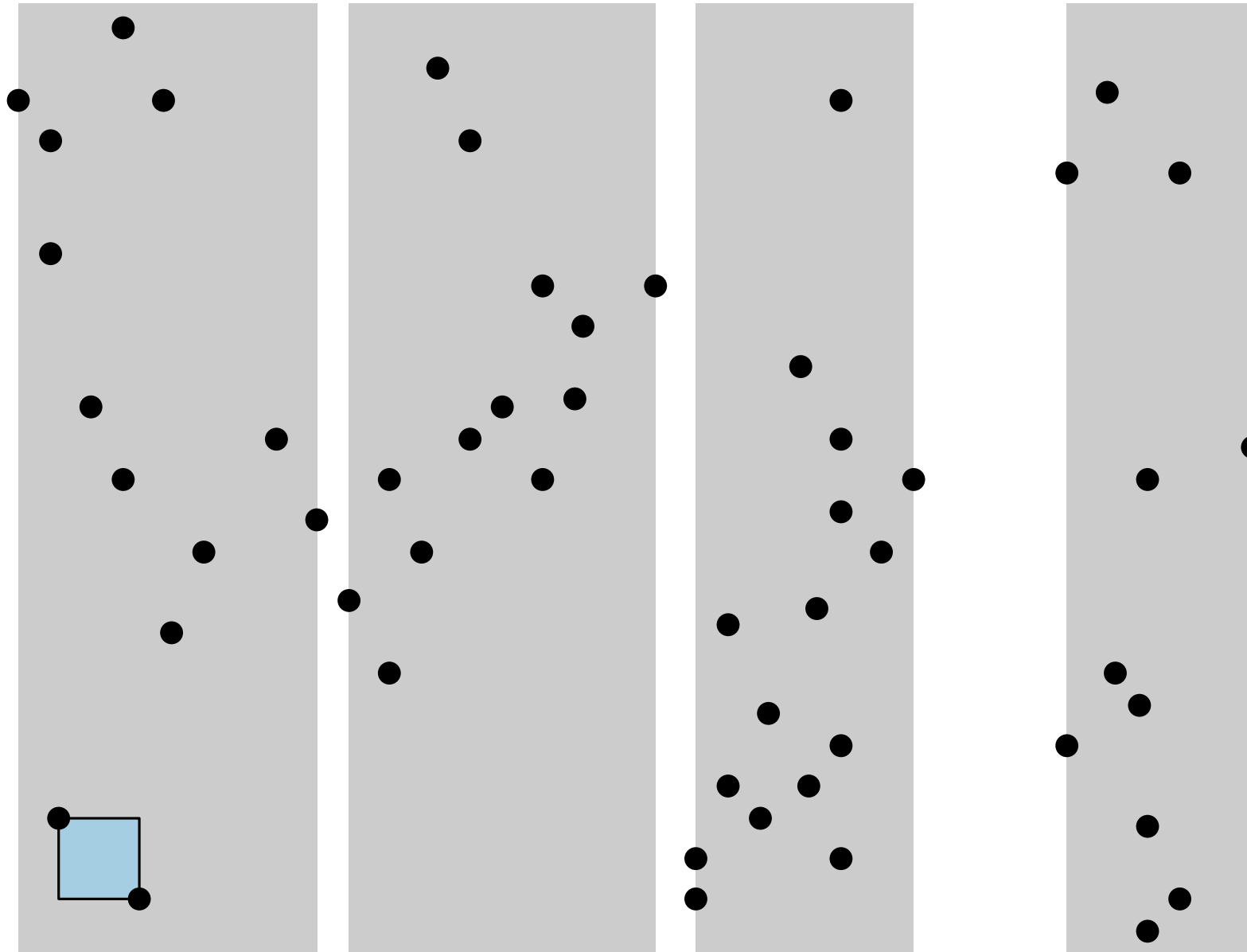
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

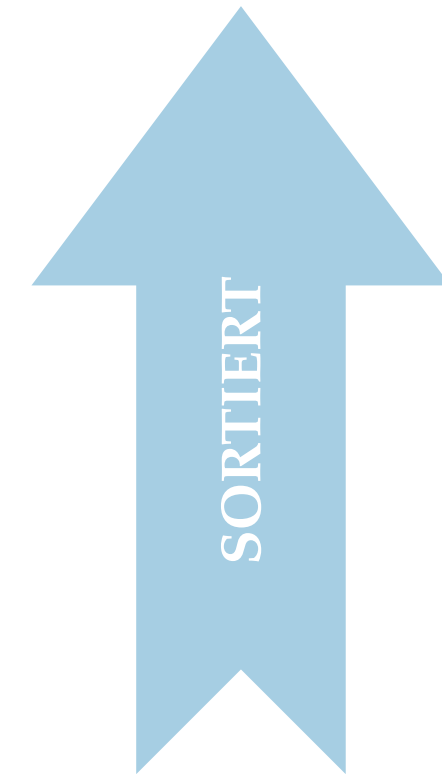
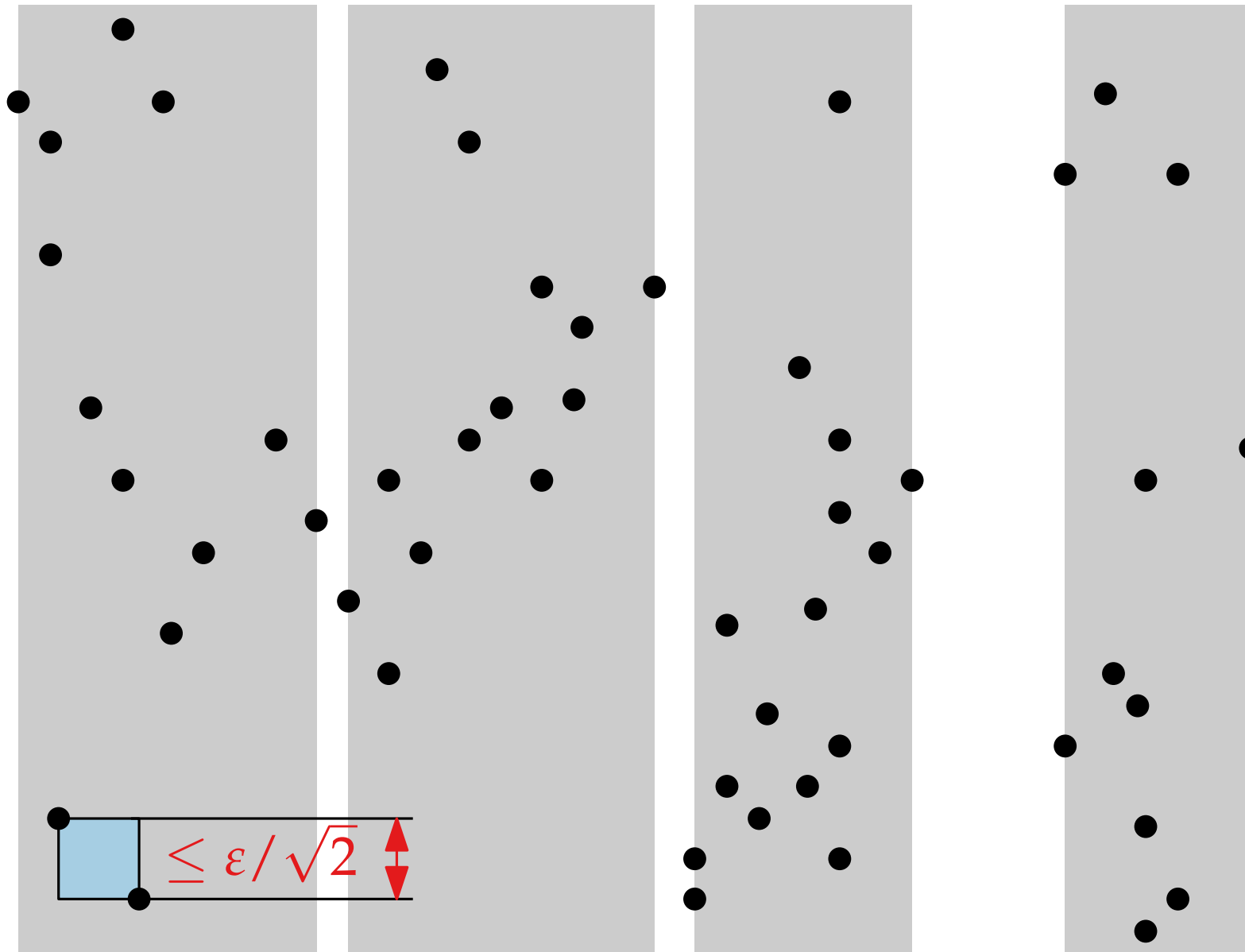
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

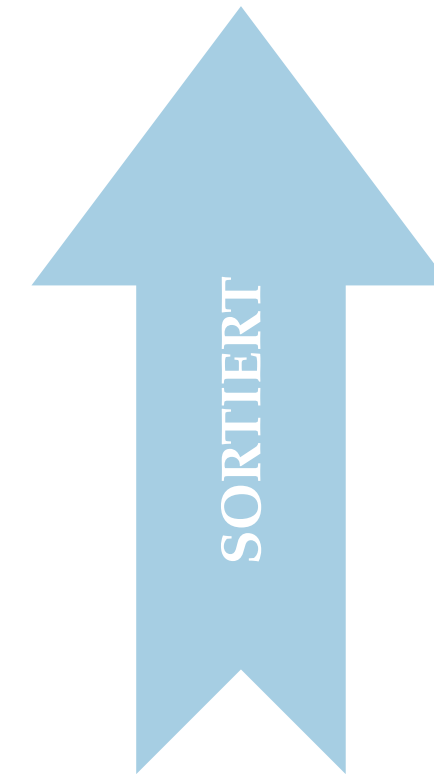
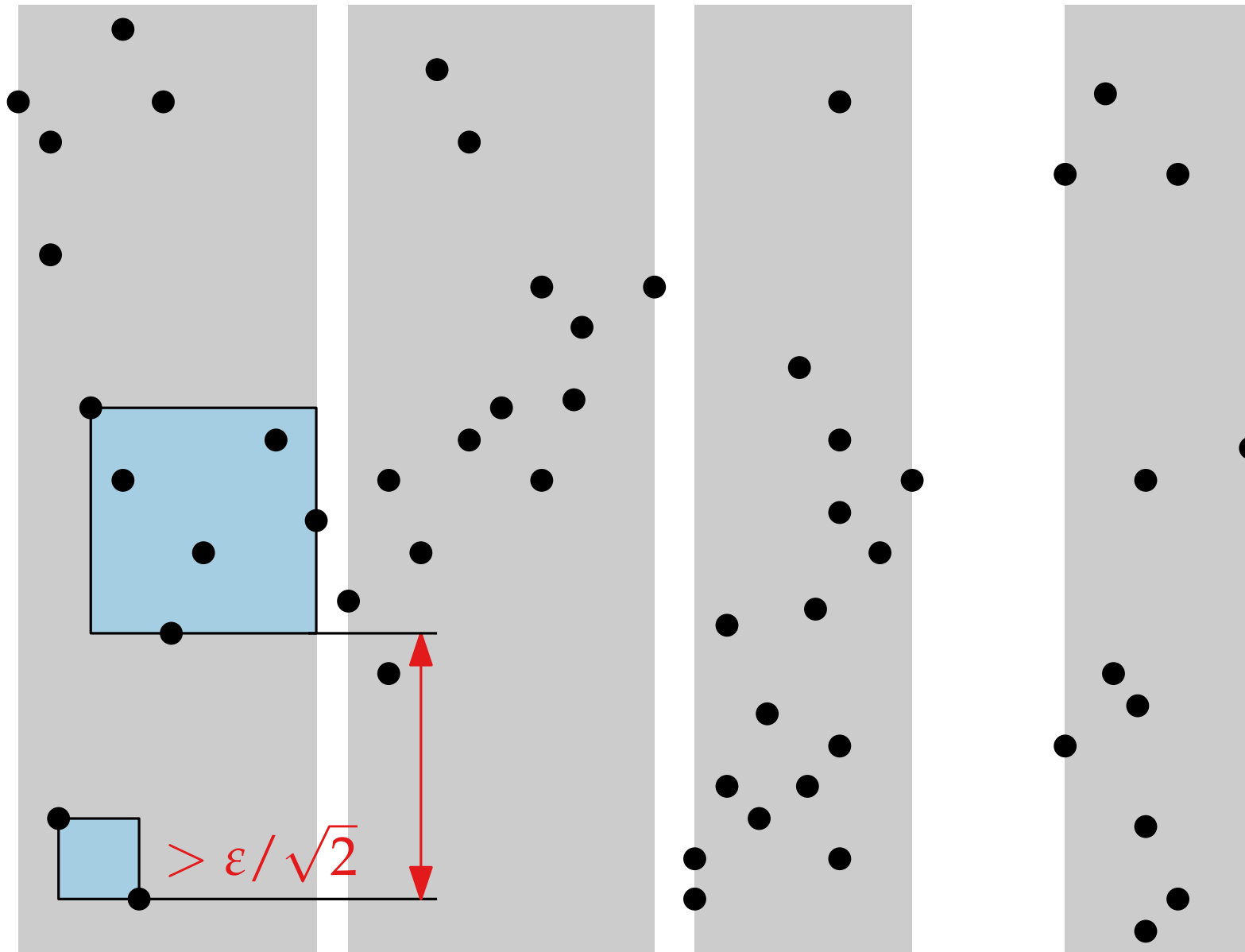
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

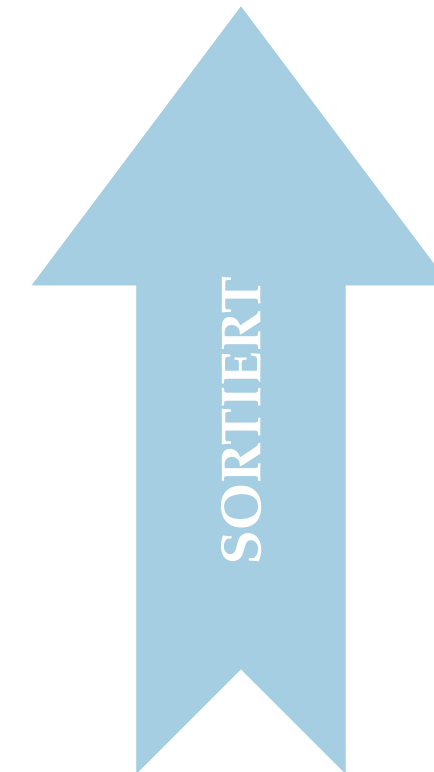
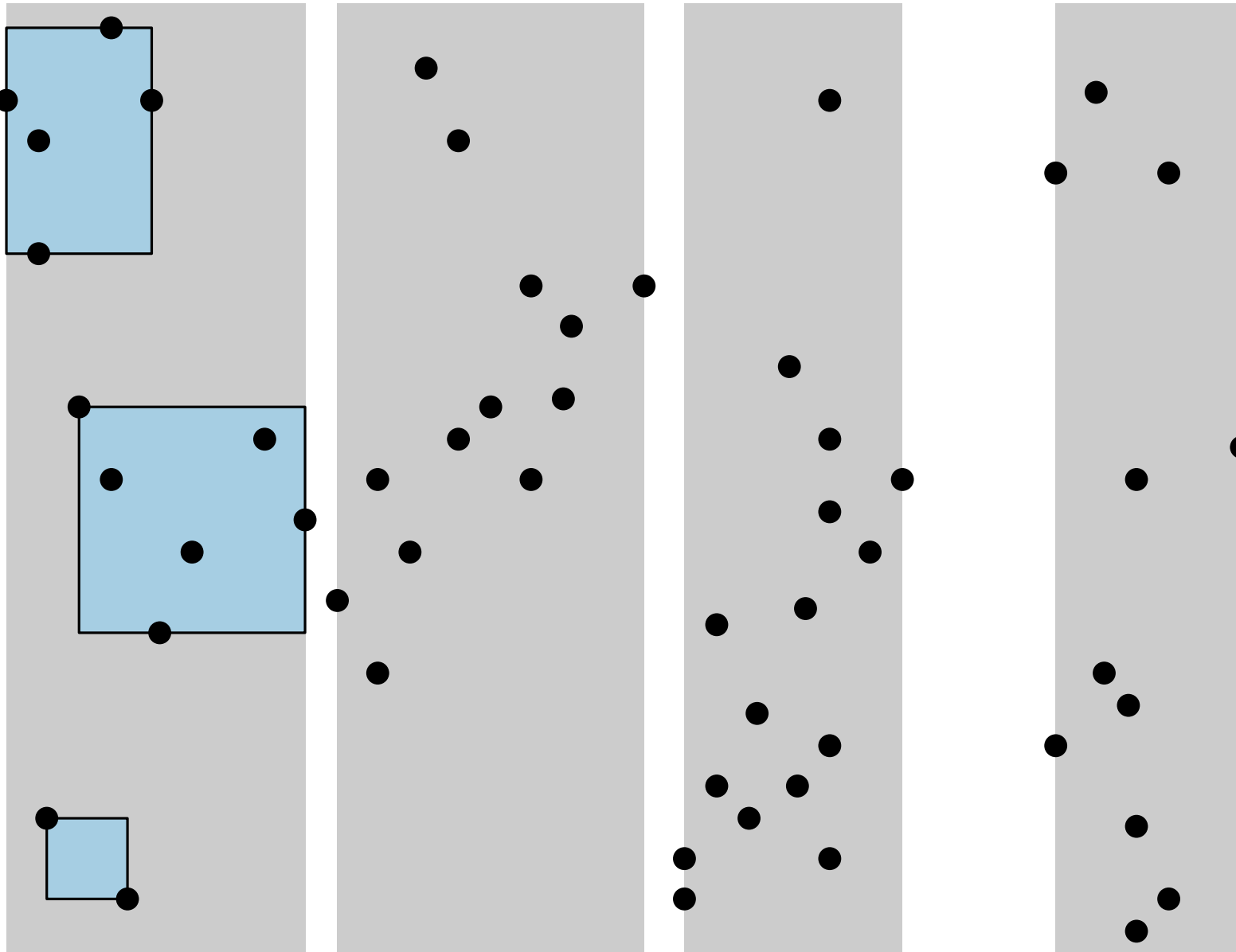
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

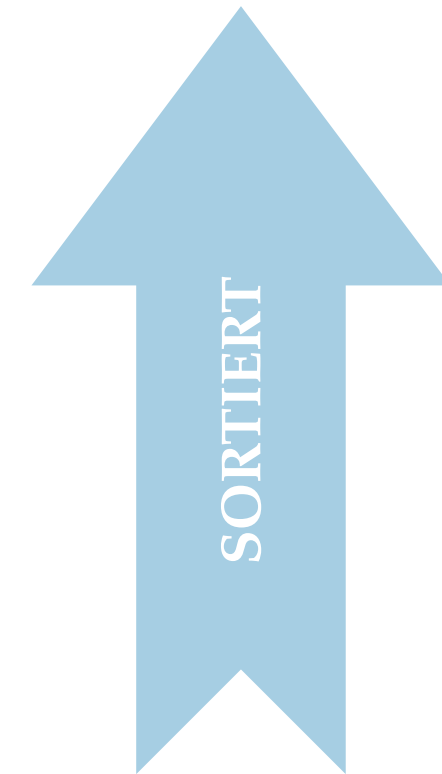
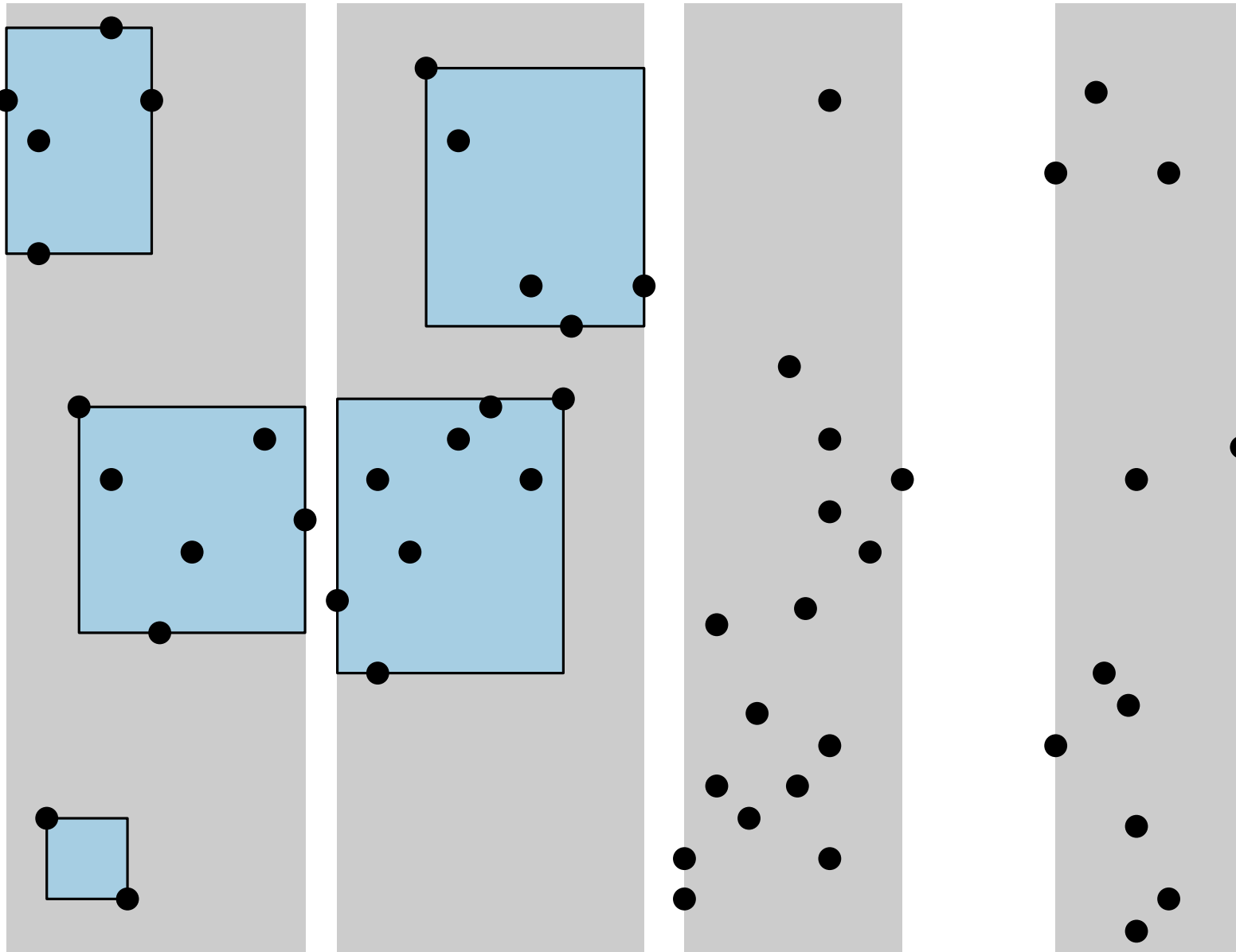
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

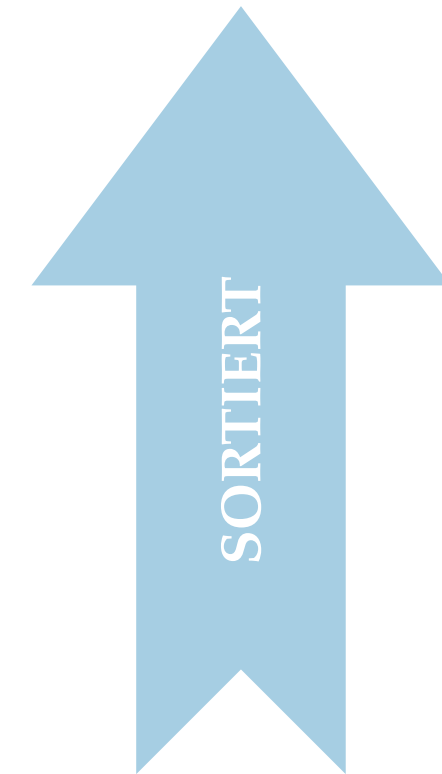
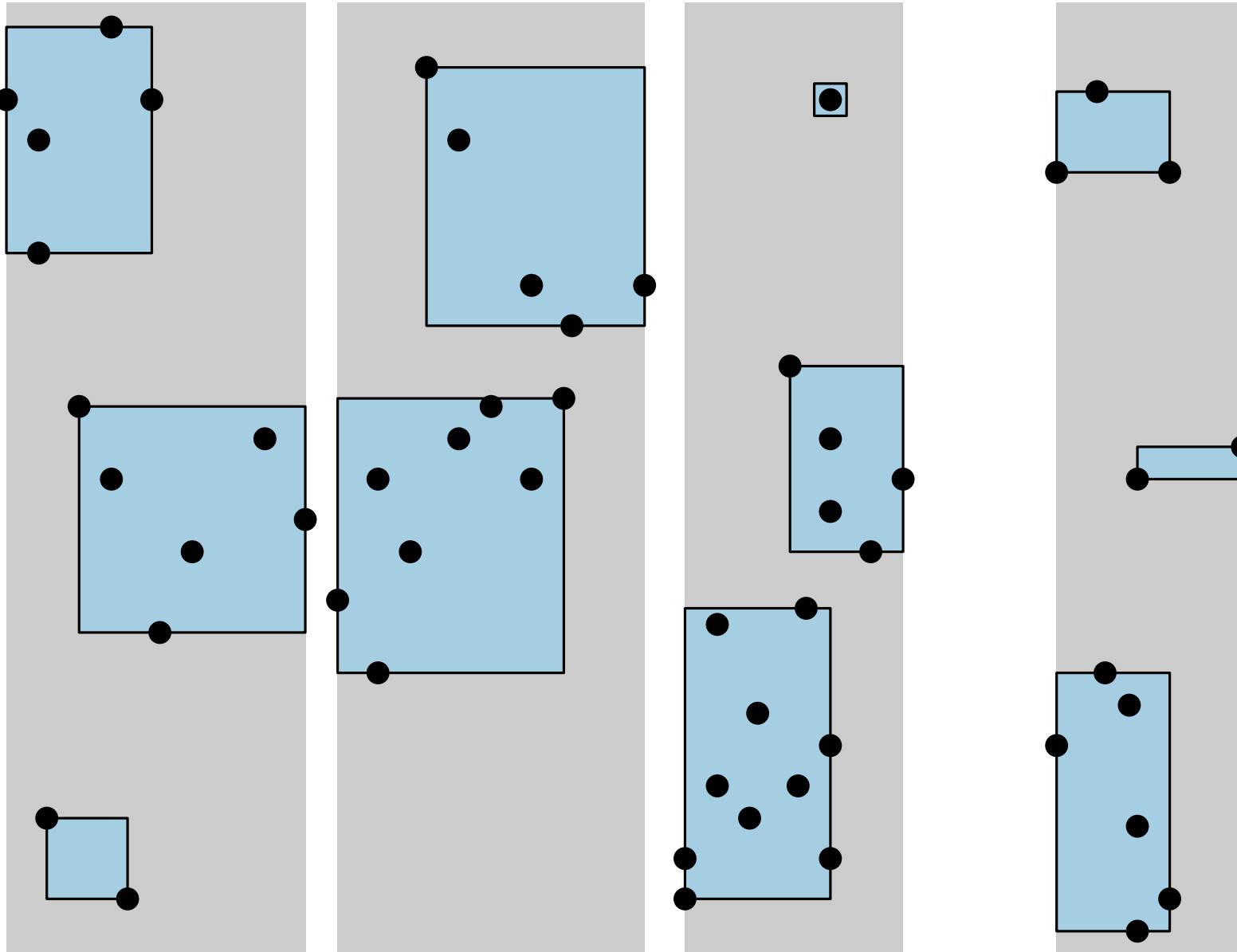
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$

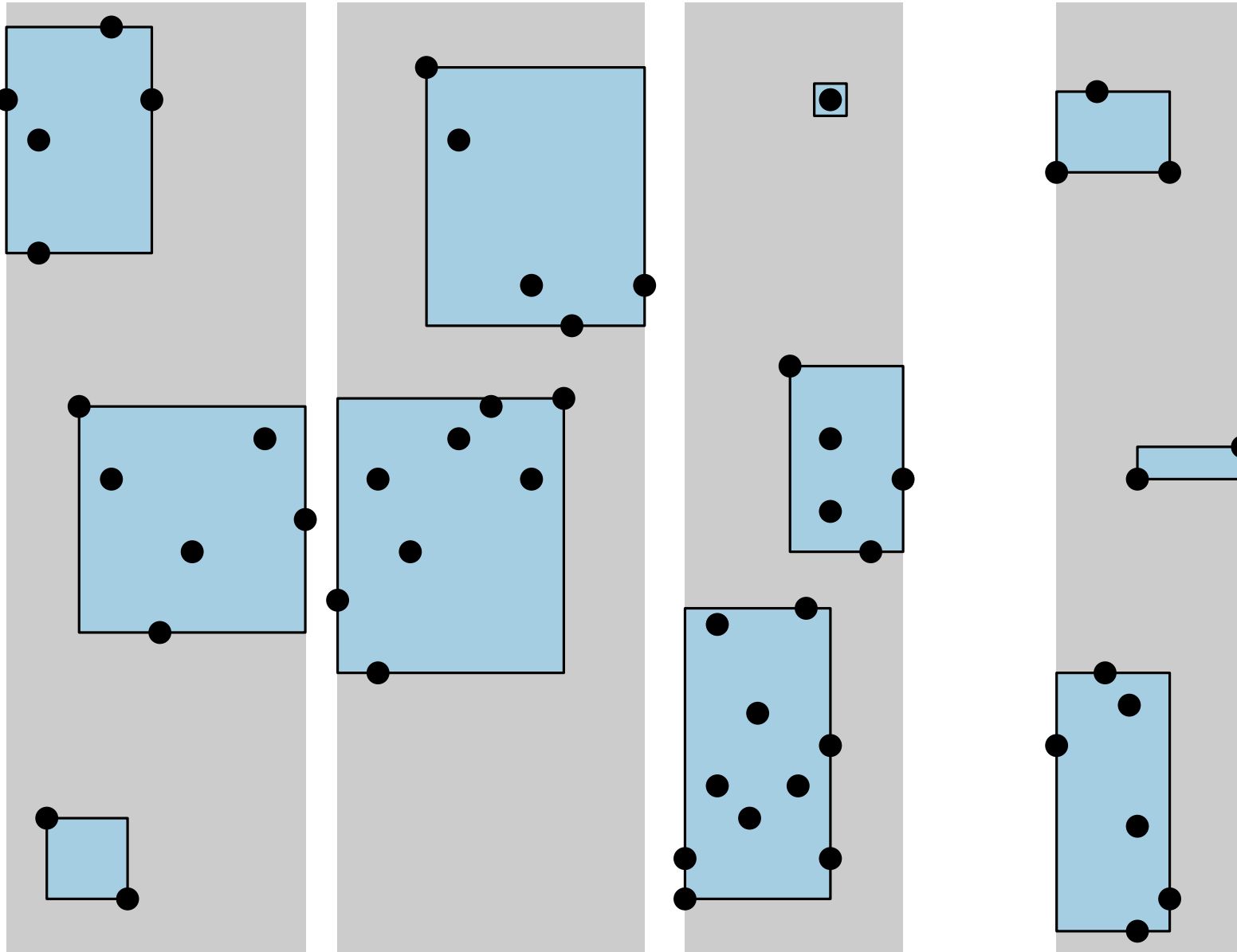


1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

- Nach x sortieren:



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$

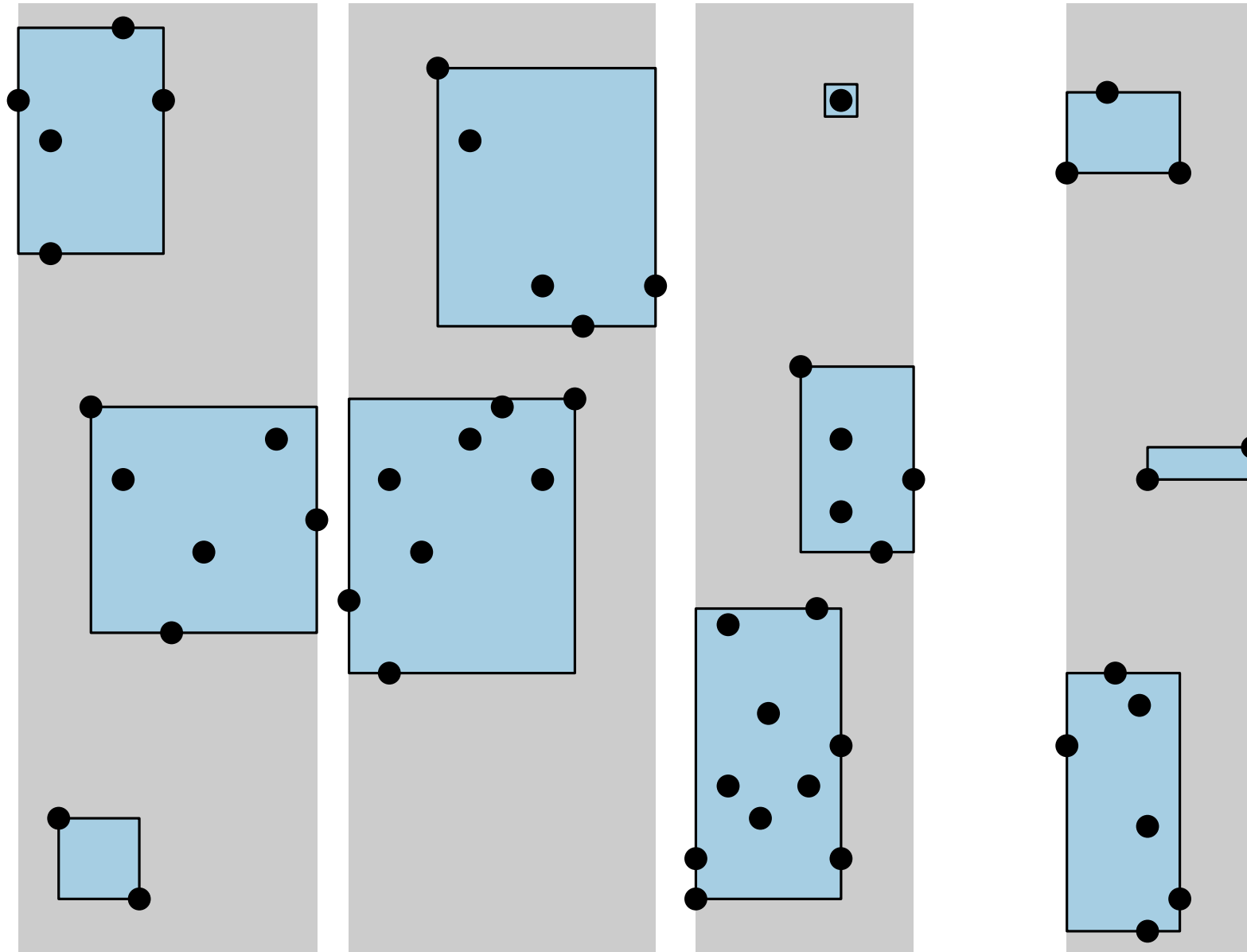


1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

- Nach x sortieren:
 $\mathcal{O}(n \log n)$



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$

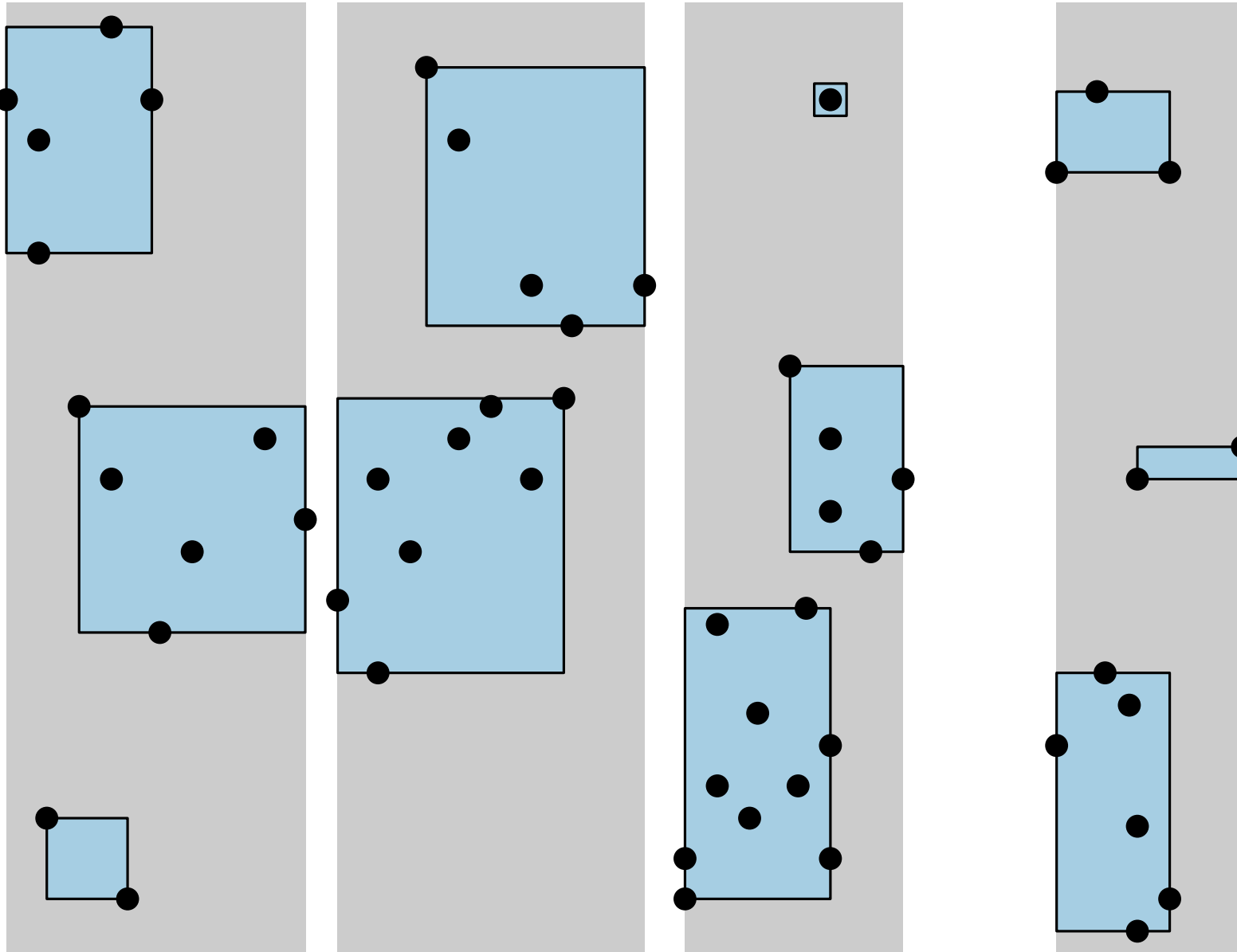


1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

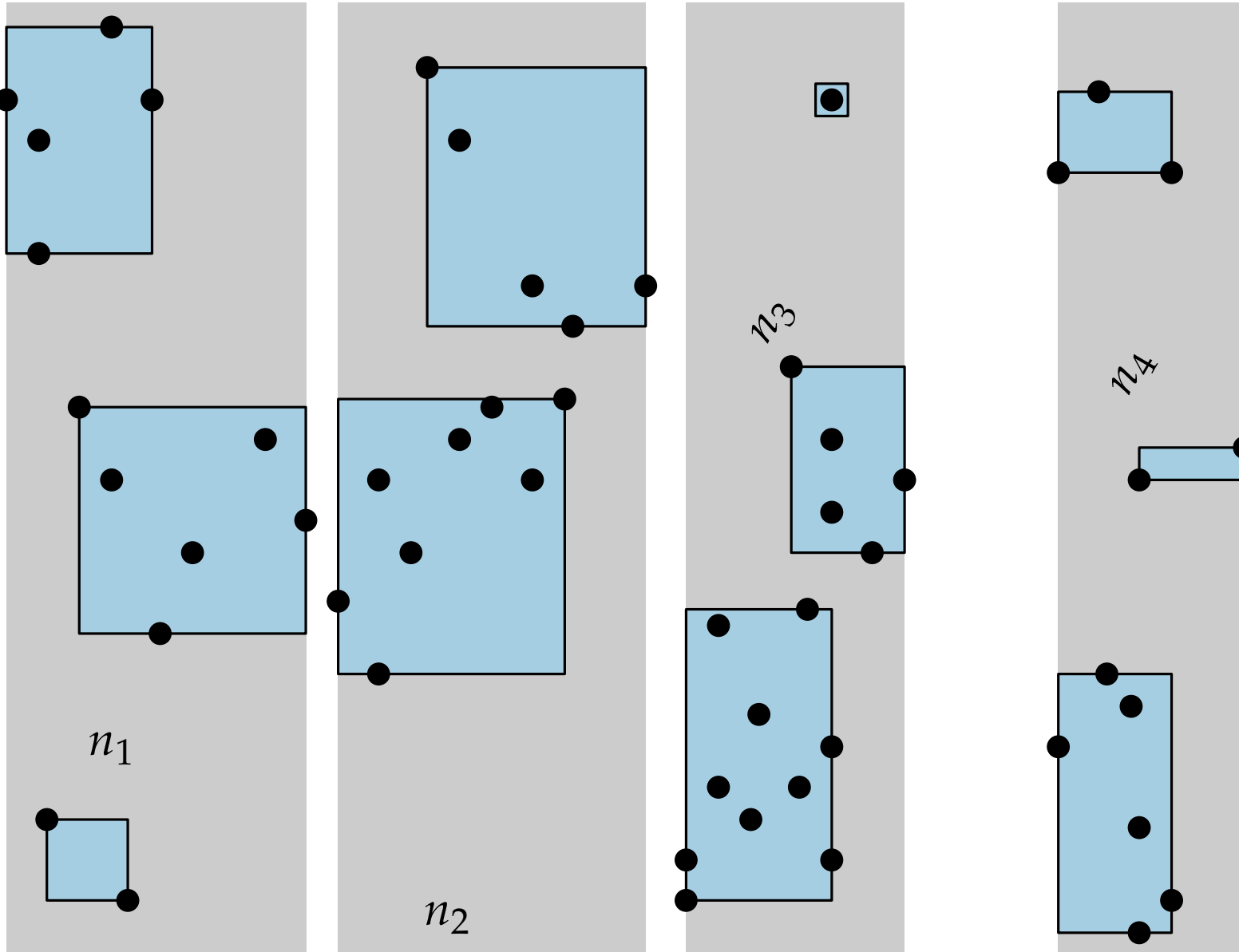
- Nach x sortieren:
 $\mathcal{O}(n \log n)$
- Pro Streifen nach y sortieren:



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

- Nach x sortieren:
 $\mathcal{O}(n \log n)$
- Pro Streifen nach y sortieren:

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$

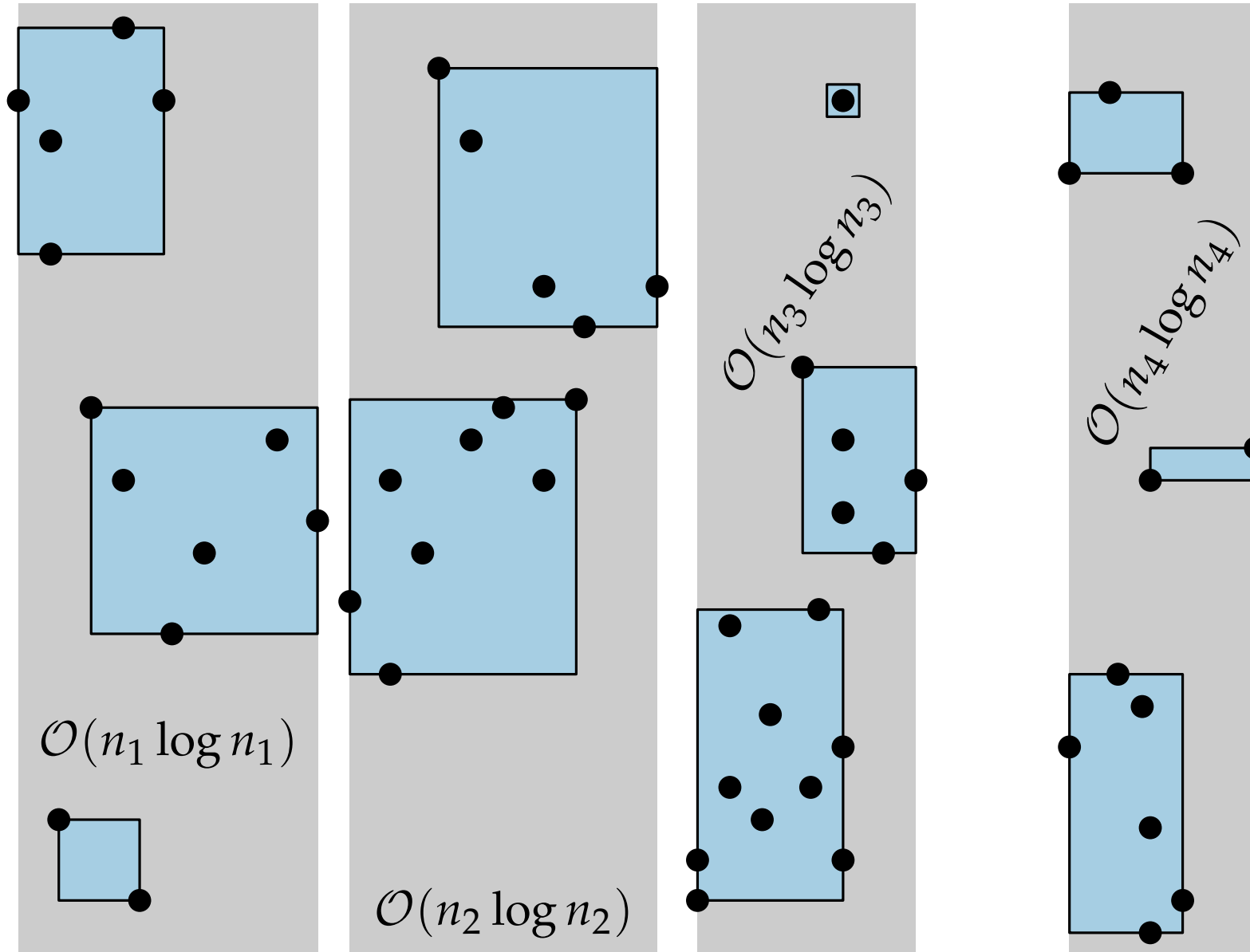


1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

- Nach x sortieren:
 $\mathcal{O}(n \log n)$
- Pro Streifen nach y sortieren:



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$

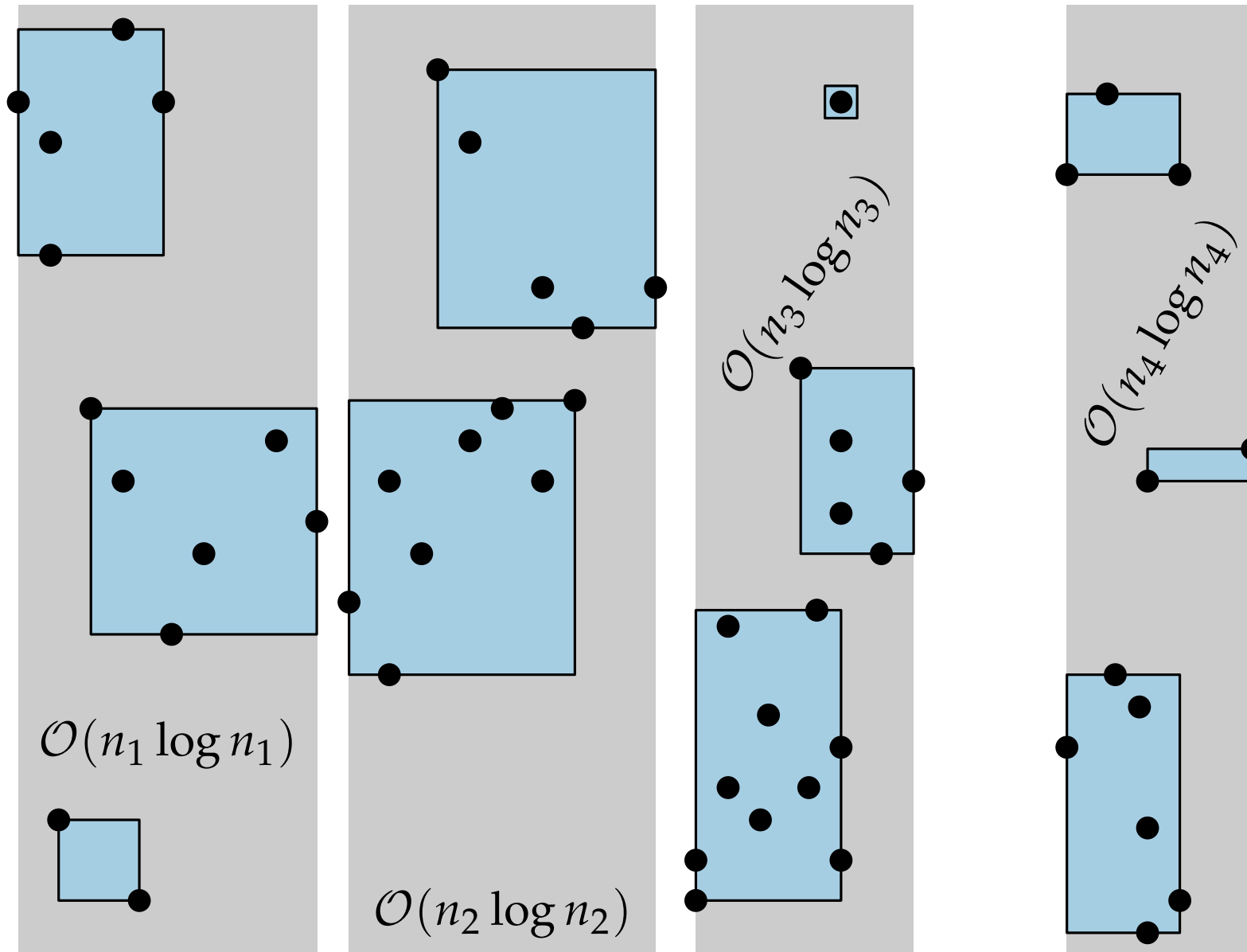


1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

- Nach x sortieren:
 $\mathcal{O}(n \log n)$
- Pro Streifen nach y sortieren:
 $\sum_j \mathcal{O}(n_j \log n_j)$



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$

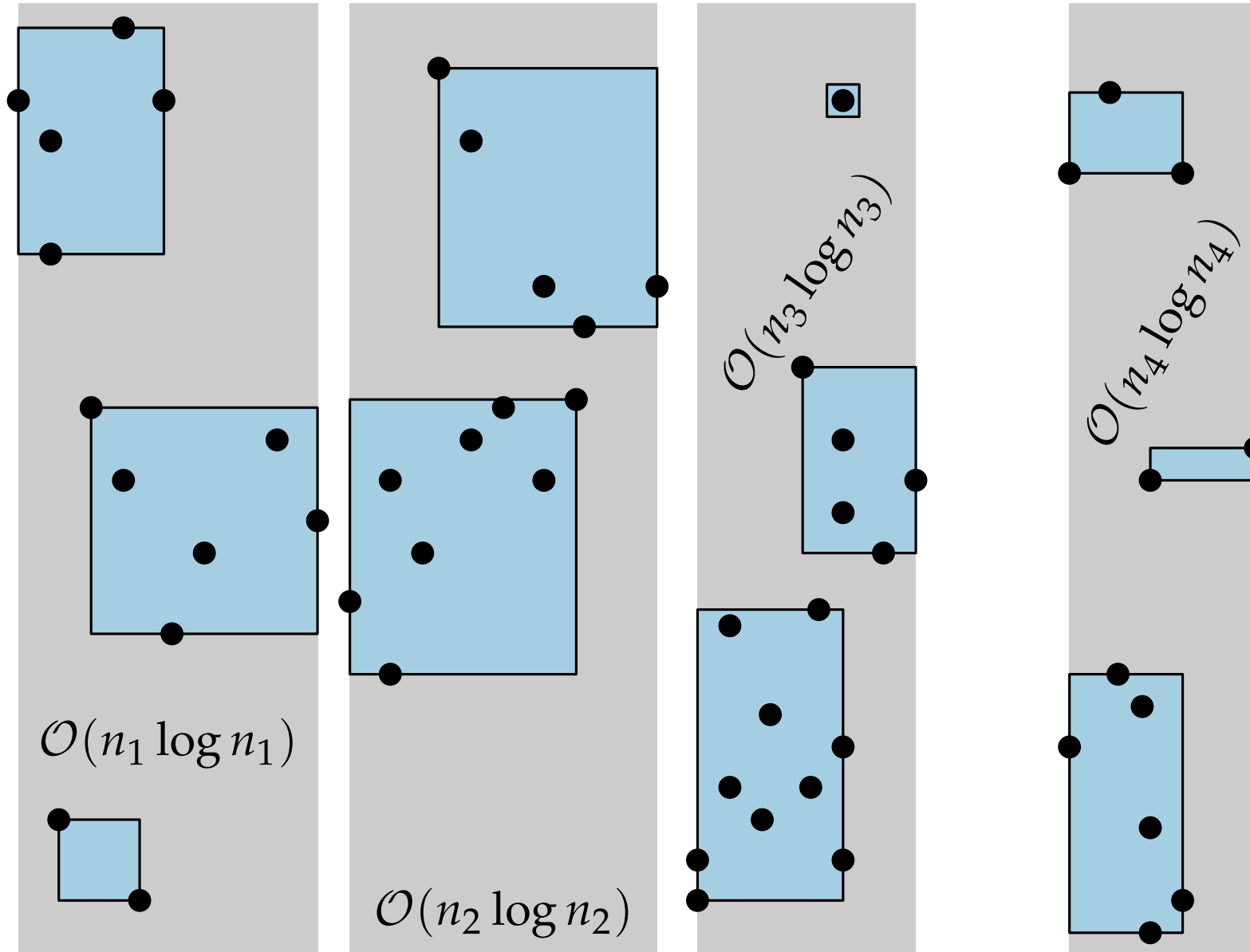


1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

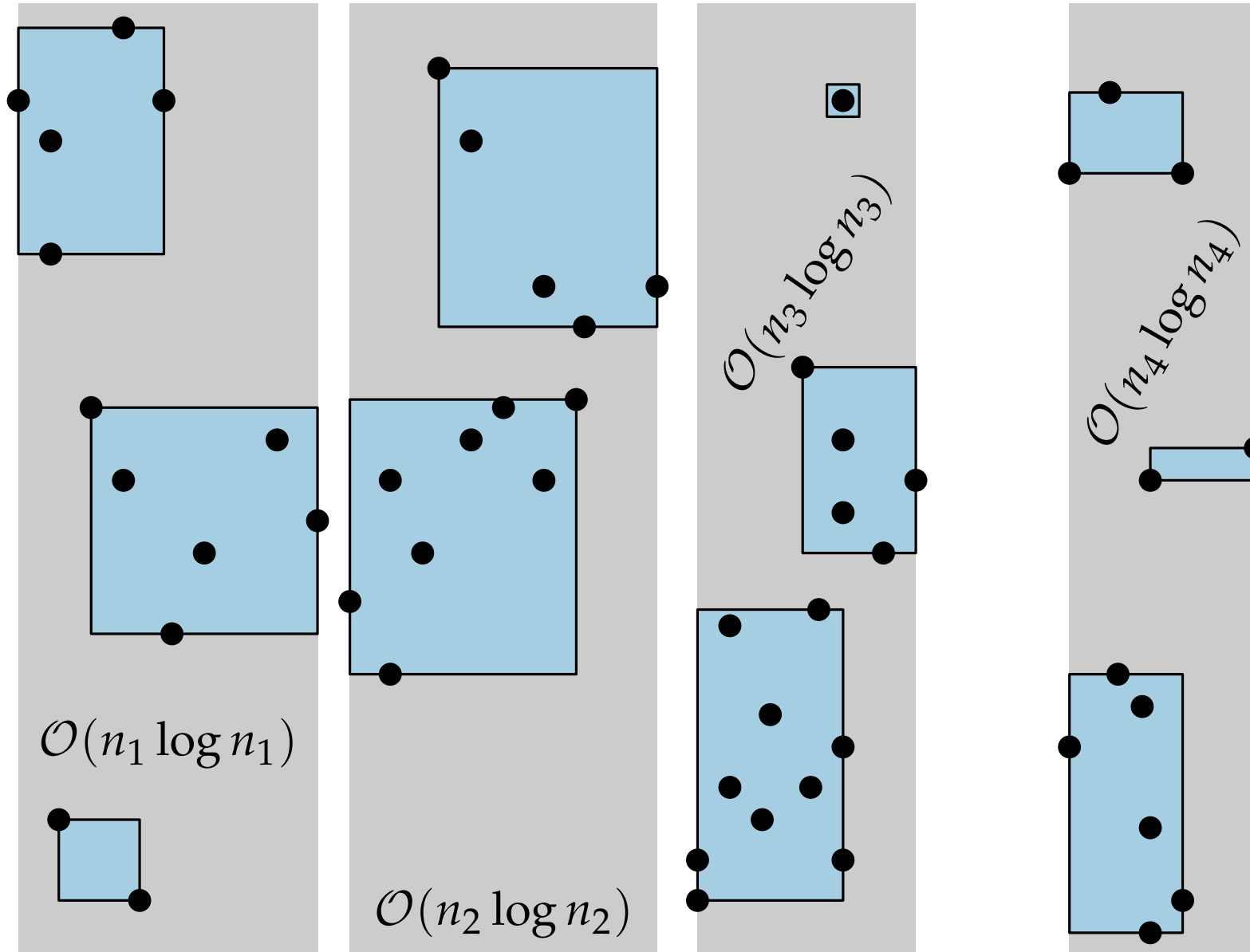
- Nach x sortieren:
 $\mathcal{O}(n \log n)$
- Pro Streifen nach y sortieren:
 $\sum_j \mathcal{O}(n_j \log n_j) = \mathcal{O}(n \log n)$



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



1. Erstelle Boxen.

- Füge Punkte hinzu, solange Streifenbreite $\leq \varepsilon/\sqrt{2}$.
- Füge Punkte hinzu, solange Streifenhöhe $\leq \varepsilon/\sqrt{2}$.

Laufzeit.

- Nach x sortieren:
 $\mathcal{O}(n \log n)$
- Pro Streifen nach y sortieren:
 $\sum_j \mathcal{O}(n_j \log n_j) = \mathcal{O}(n \log n)$
- Insgesamt:
 $\mathcal{O}(n \log n)$

Box Graph $\mathcal{G}_{\text{box}}$

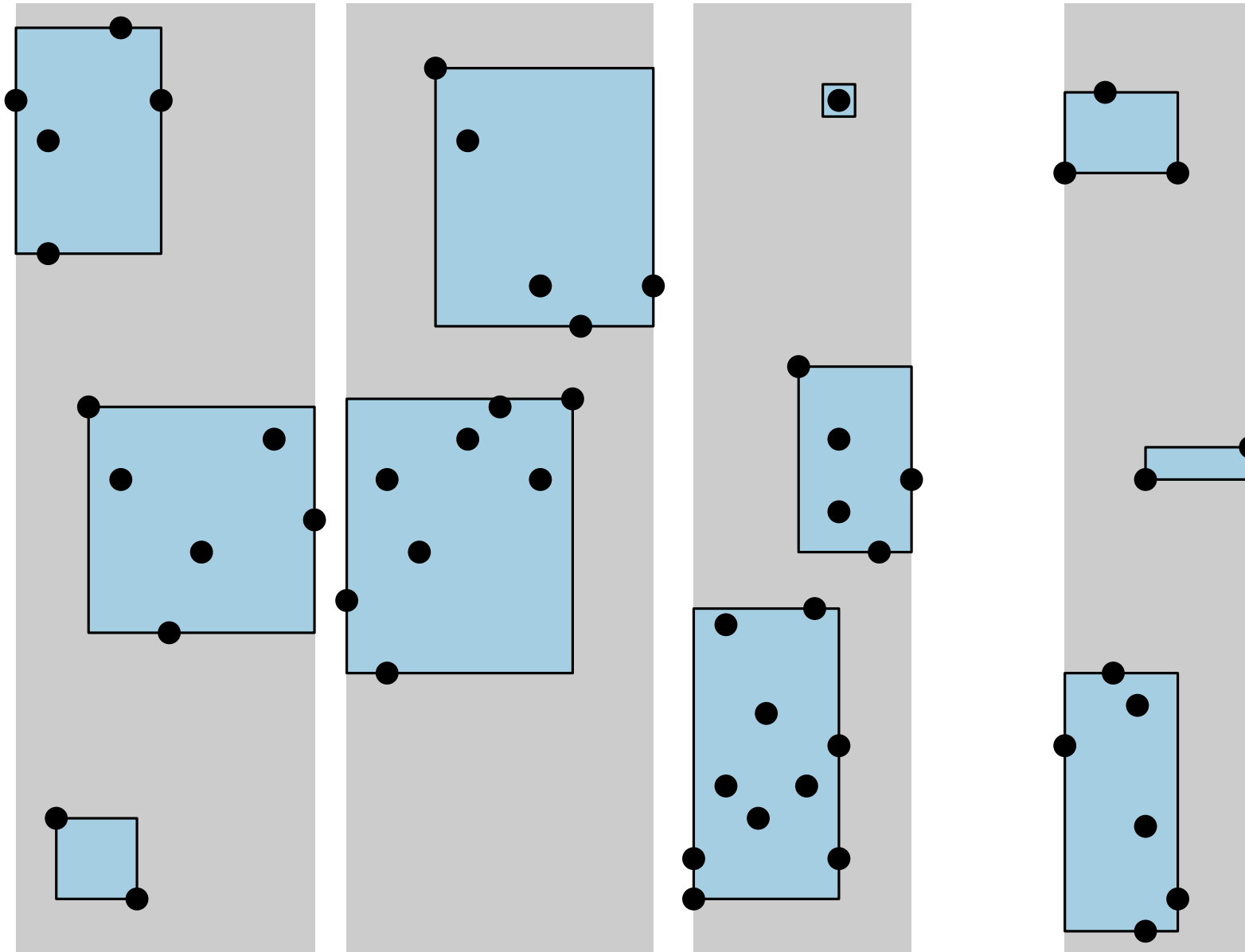
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box...



Box Graph $\mathcal{G}_{\text{box}}$

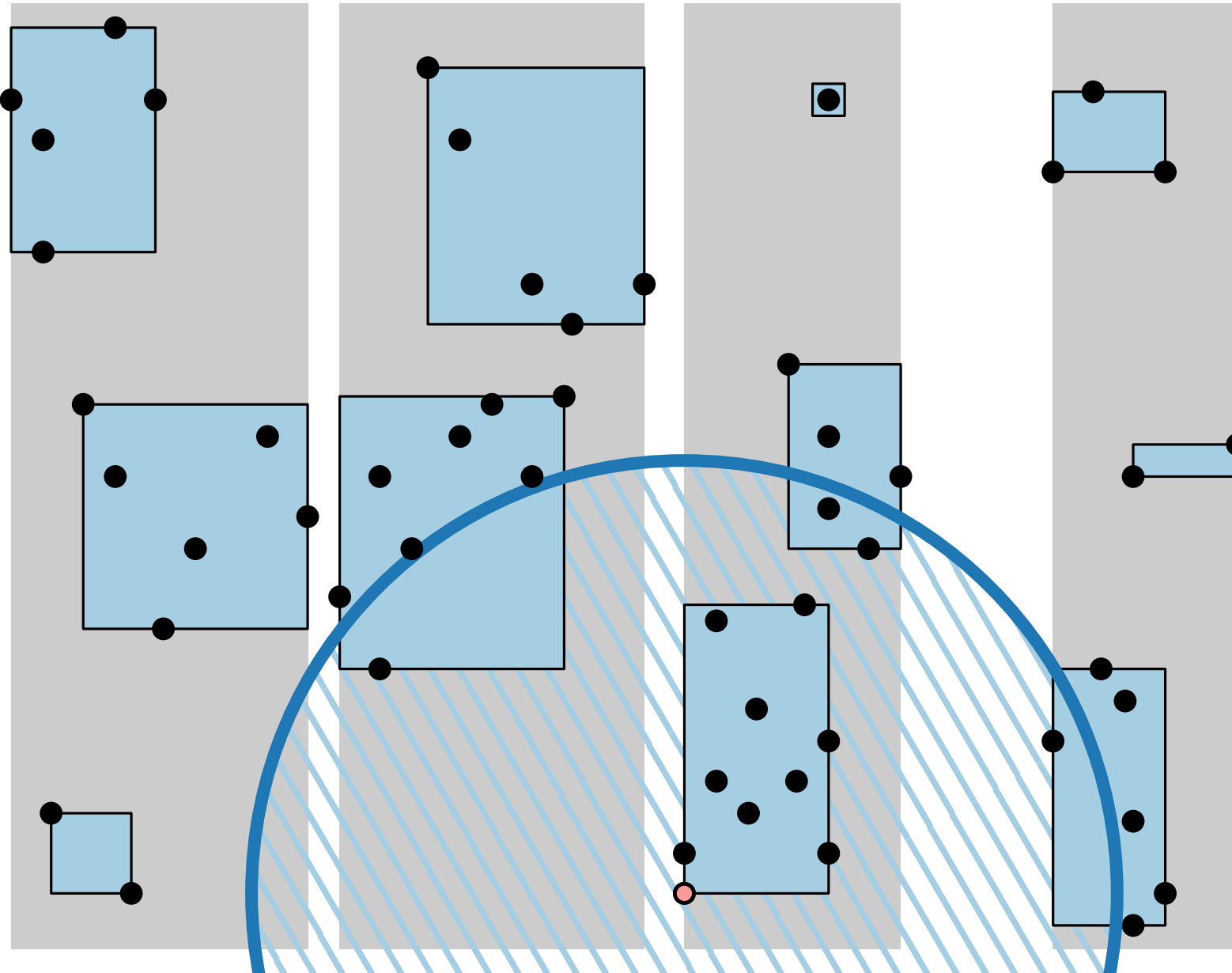
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box...



Box Graph $\mathcal{G}_{\text{box}}$

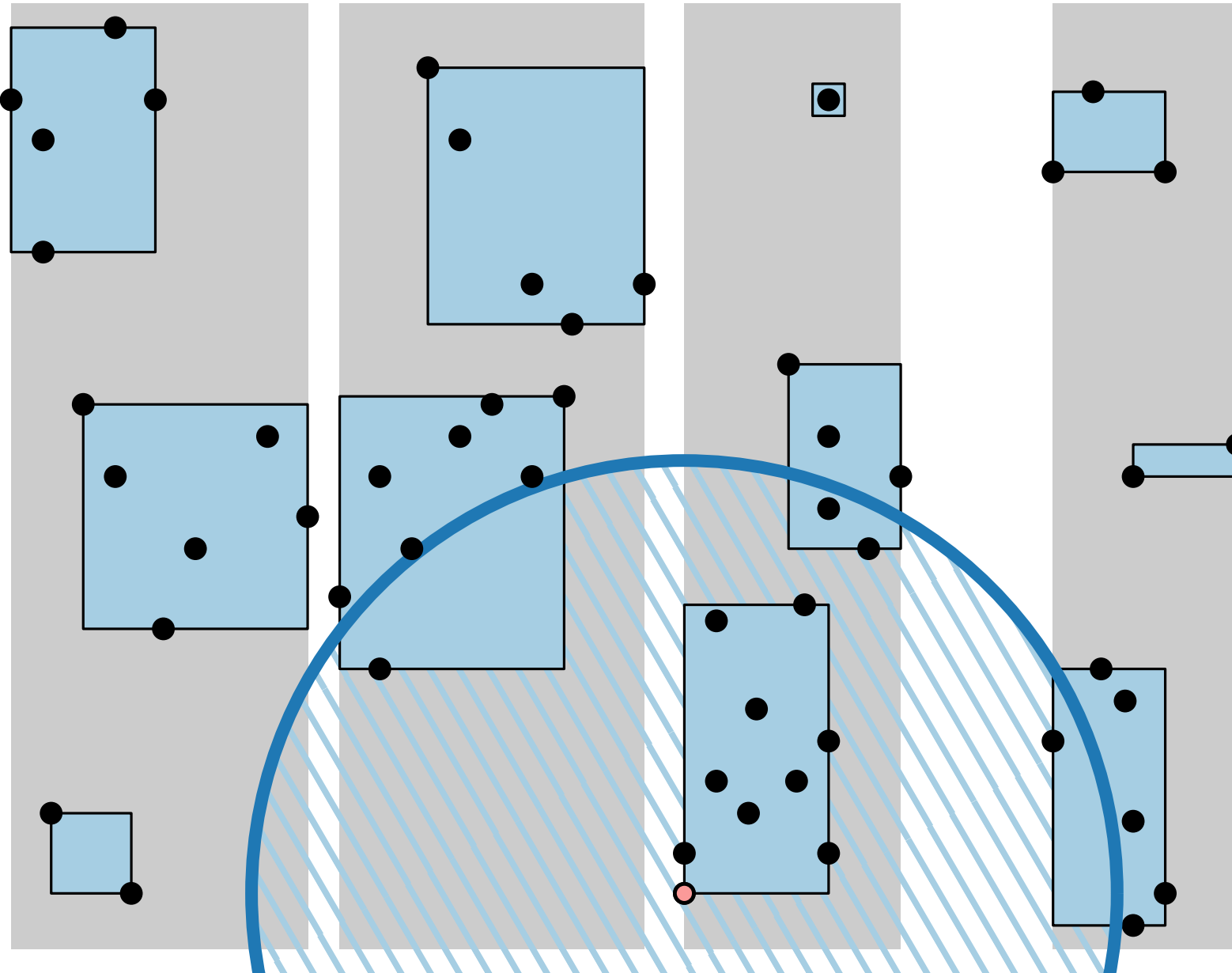
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)



Box Graph $\mathcal{G}_{\text{box}}$

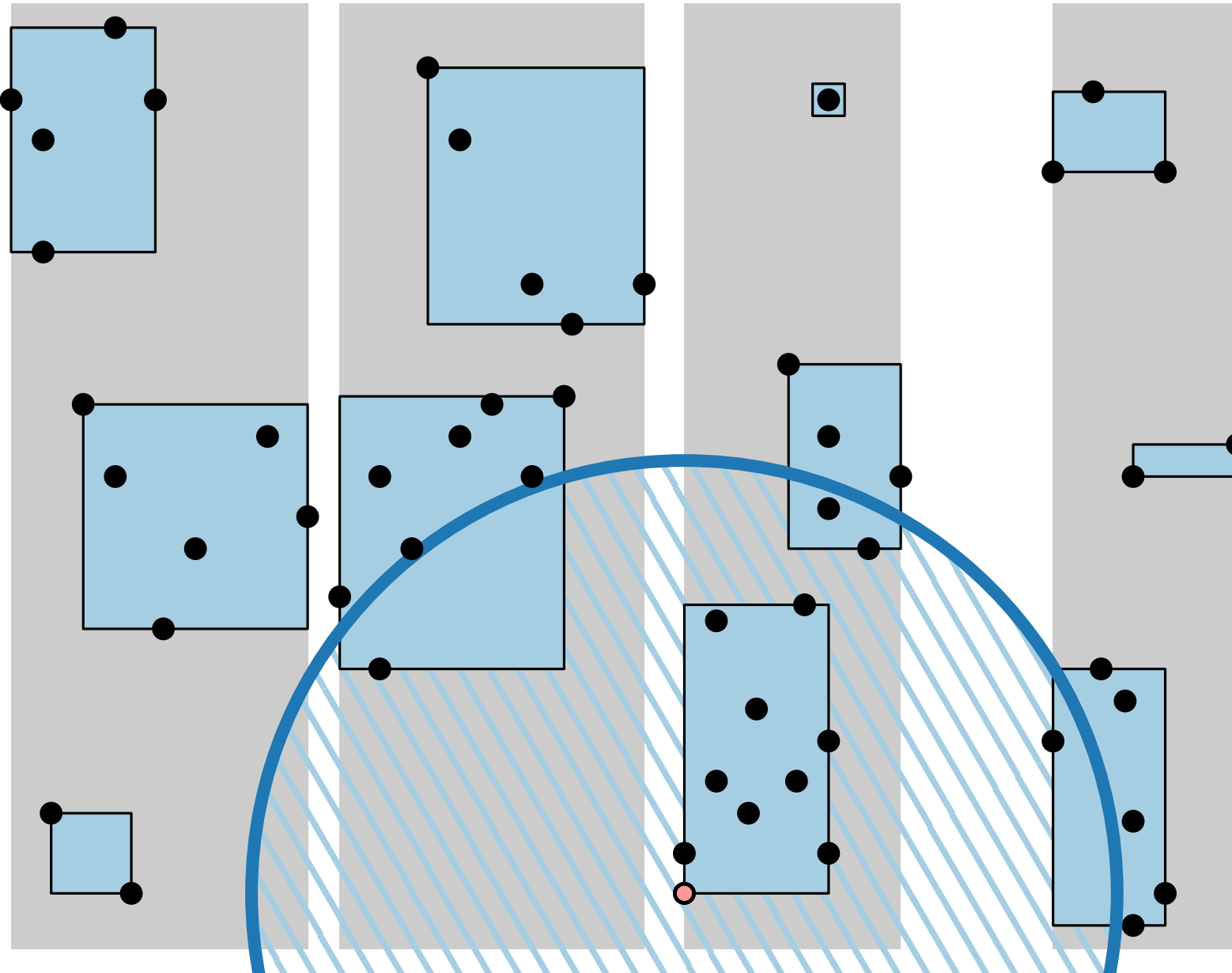
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

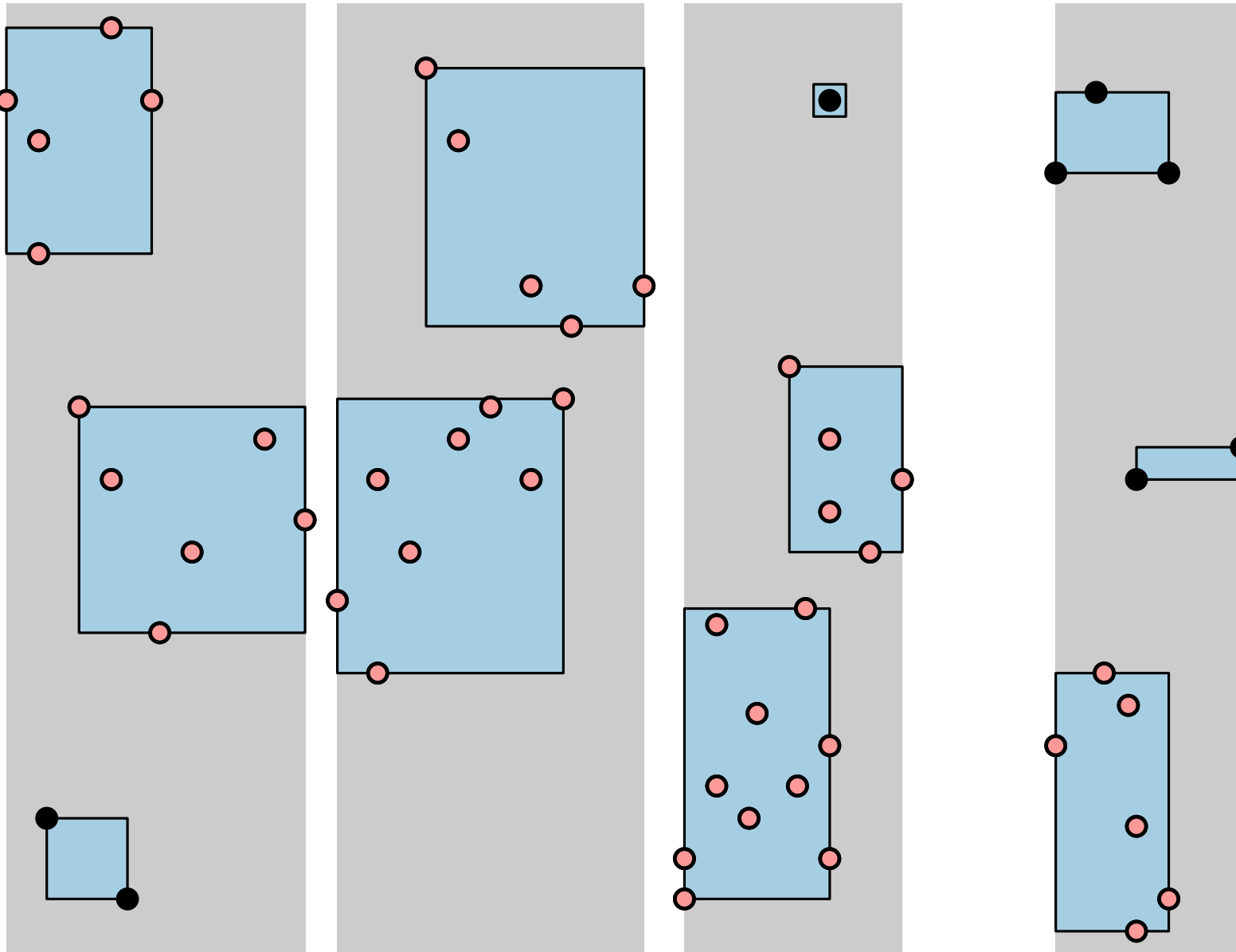
- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



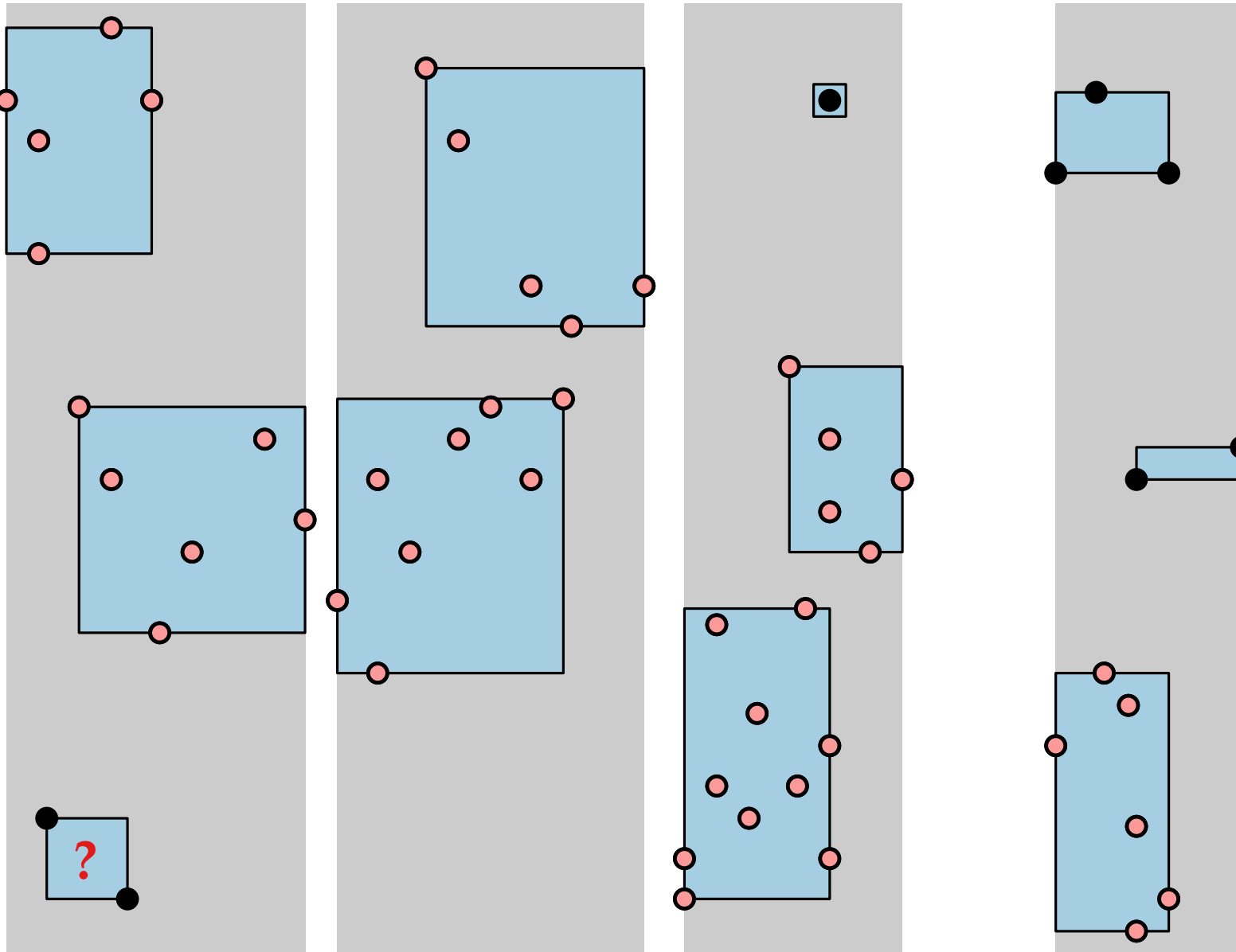
Eigenschaften der Boxen.

- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...
sind alle Punkte Kerne.

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...
sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten...

Box Graph $\mathcal{G}_{\text{box}}$

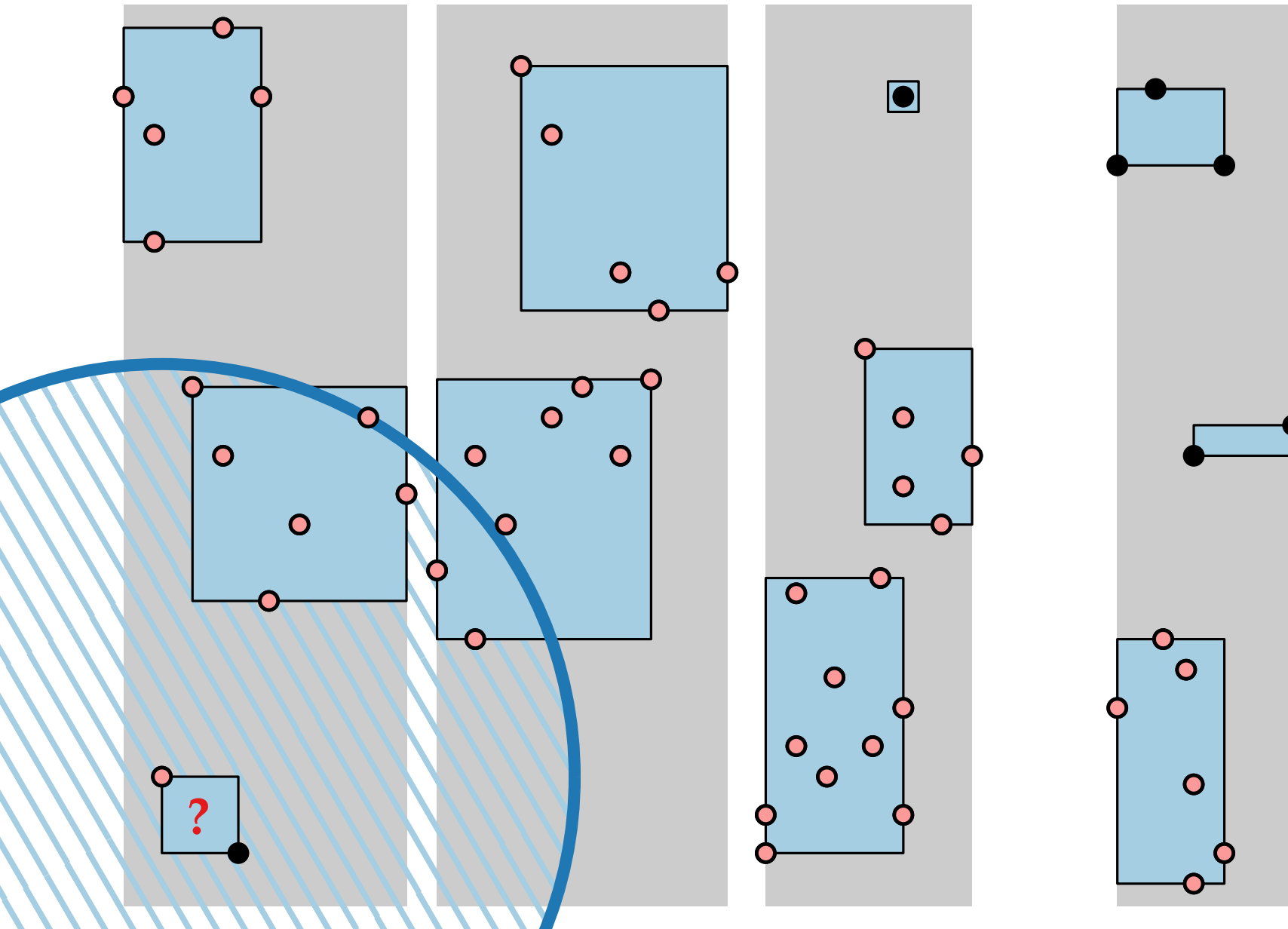
$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

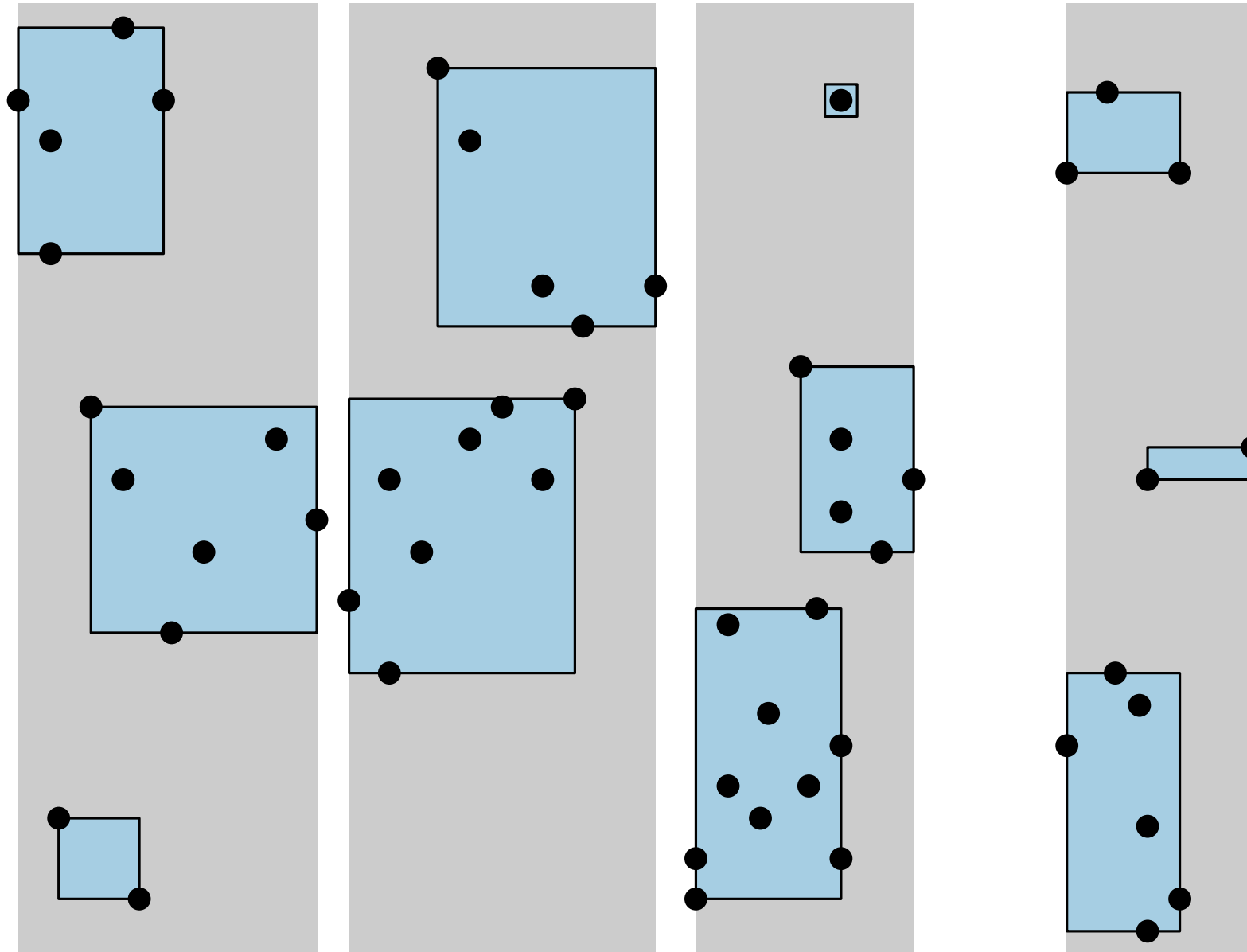
- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...
sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten...
kann es auch Kerne geben!



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...
sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten...
kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \longleftrightarrow \\ \varepsilon/\sqrt{2}: \longleftrightarrow \end{array}$$

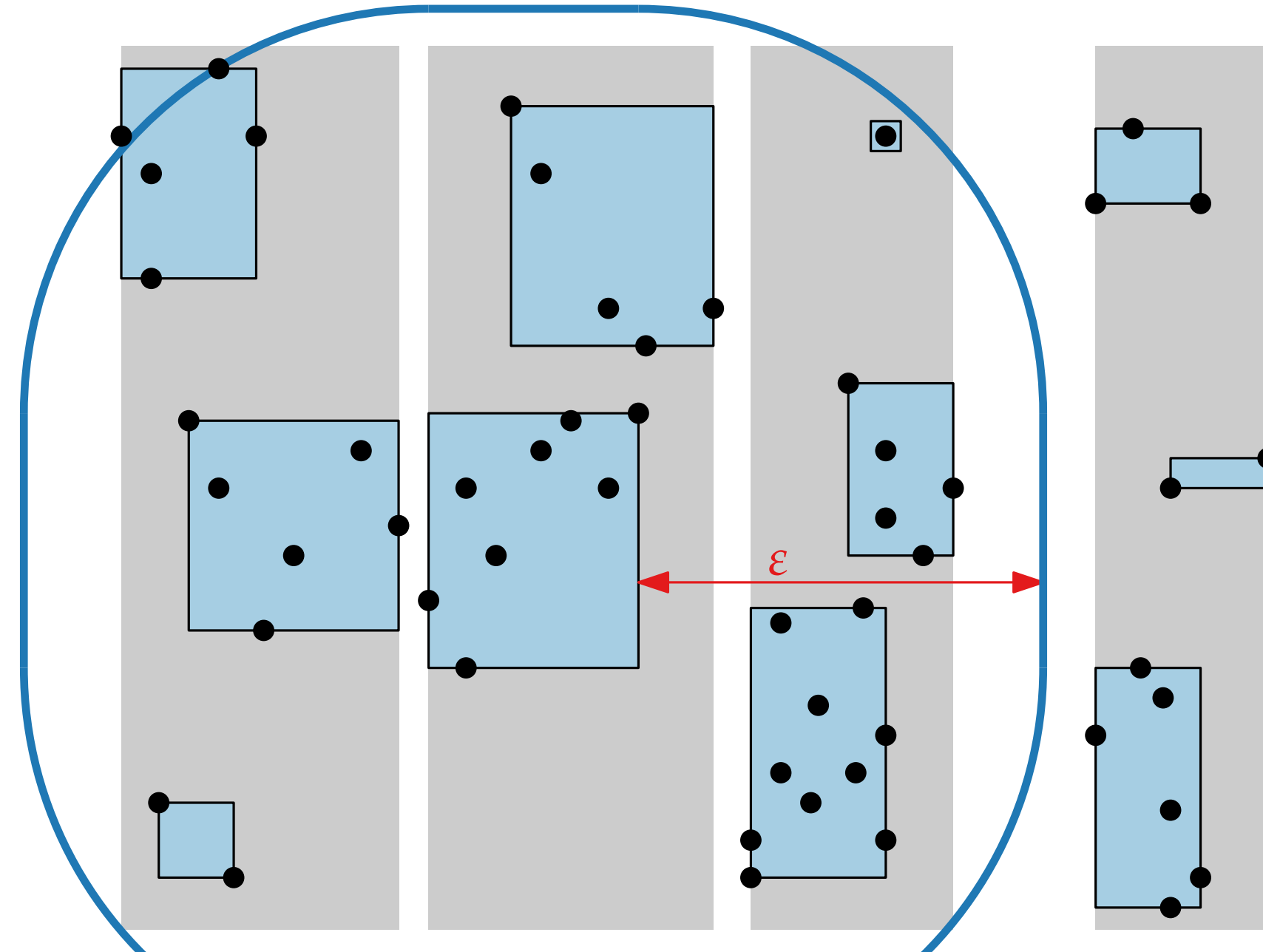


Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \longleftrightarrow \\ \varepsilon/\sqrt{2}: \longleftrightarrow \end{array}$$

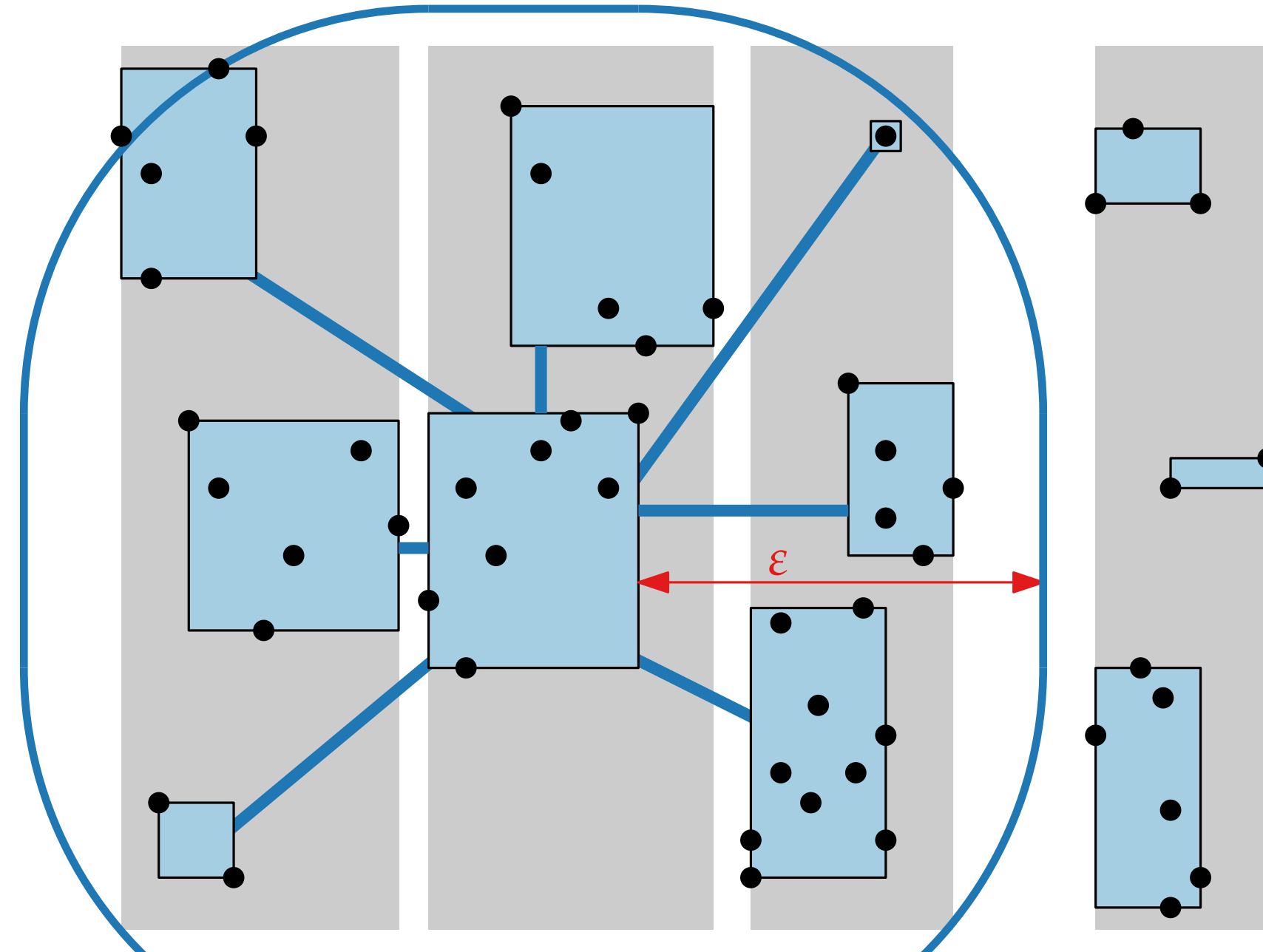


Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$

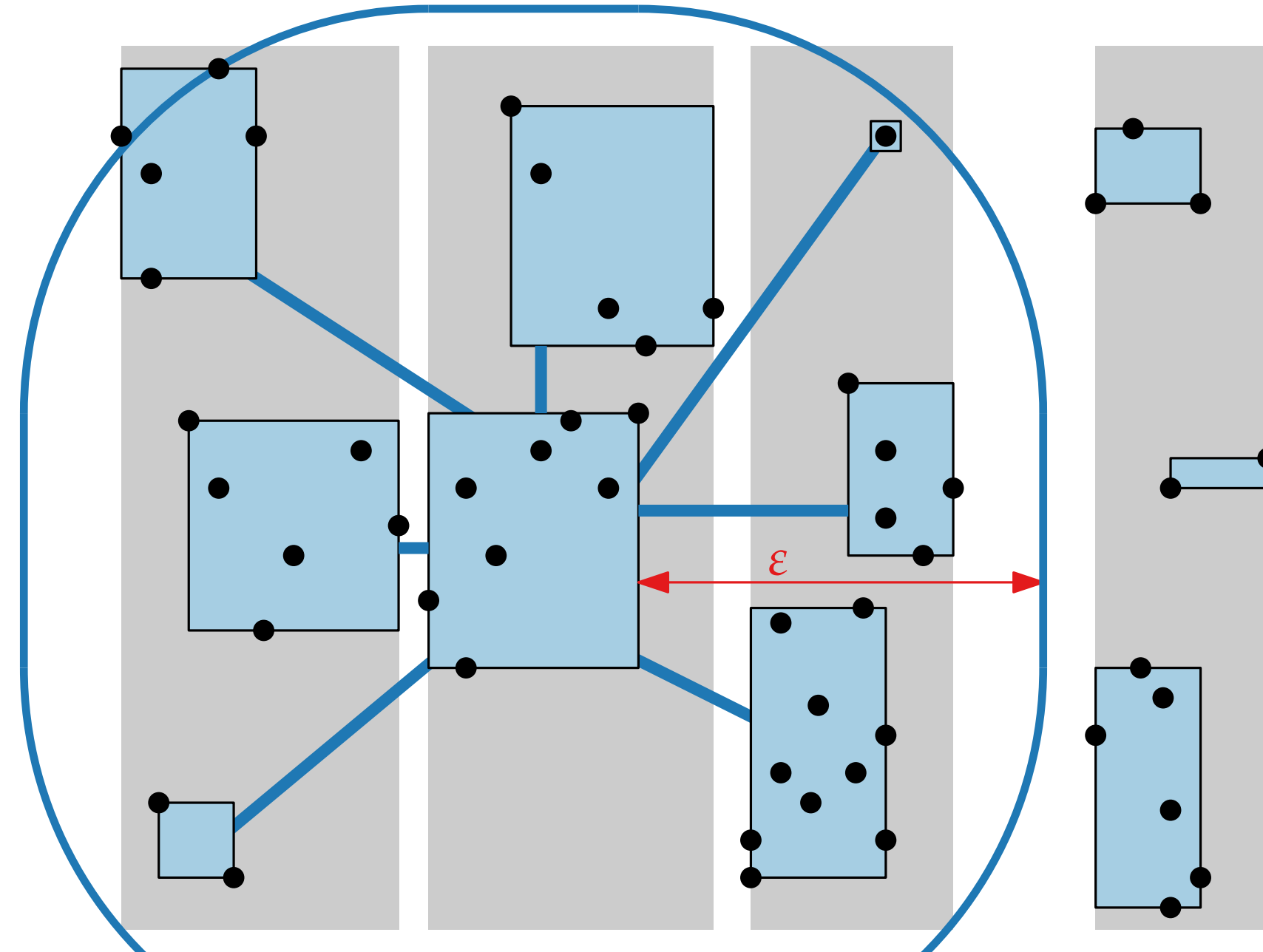


Eigenschaften der Boxen.

- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...
sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten...
kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit
Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$:



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \longleftrightarrow \\ \varepsilon/\sqrt{2}: \longleftrightarrow \end{array}$$

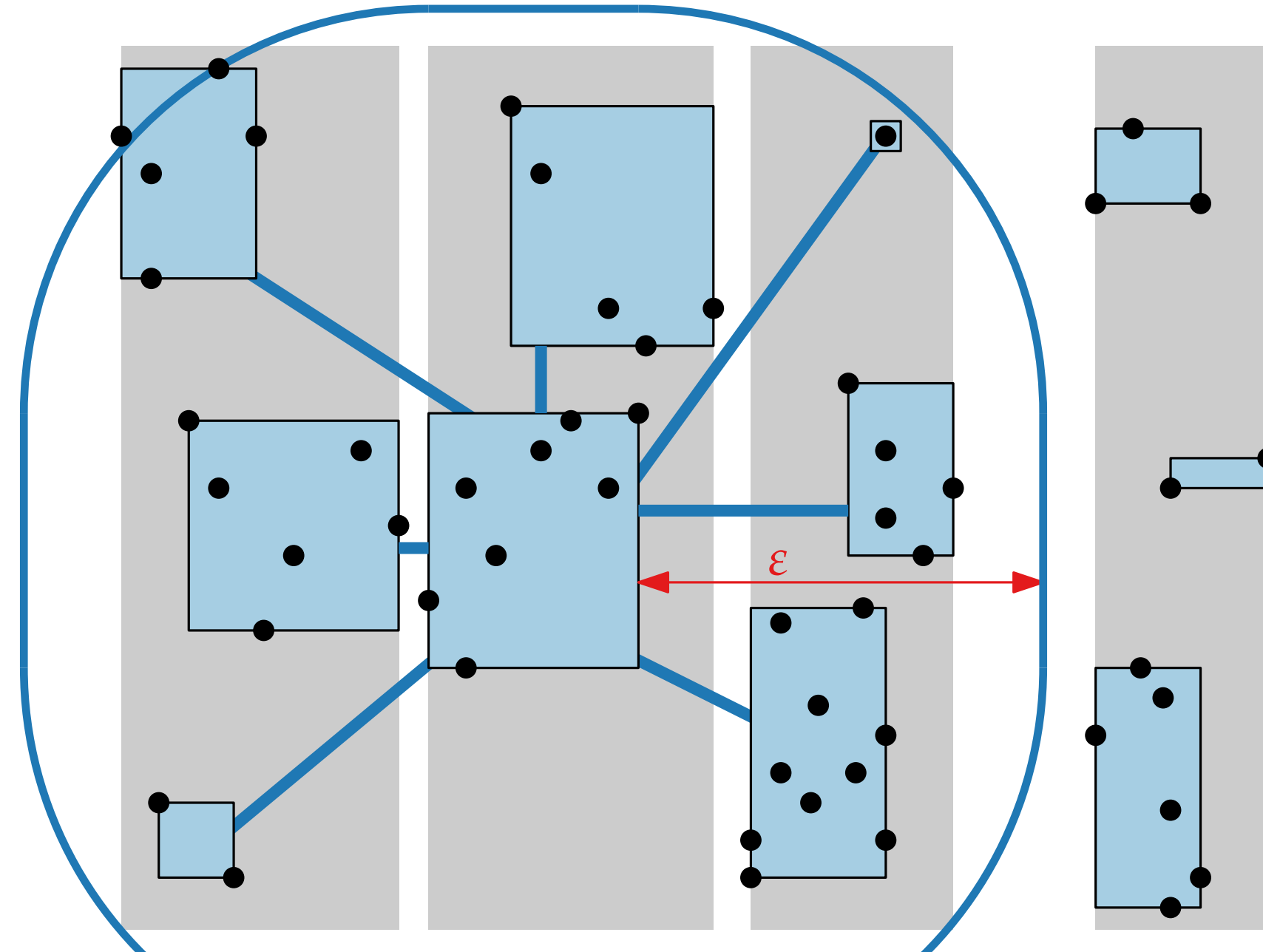


Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

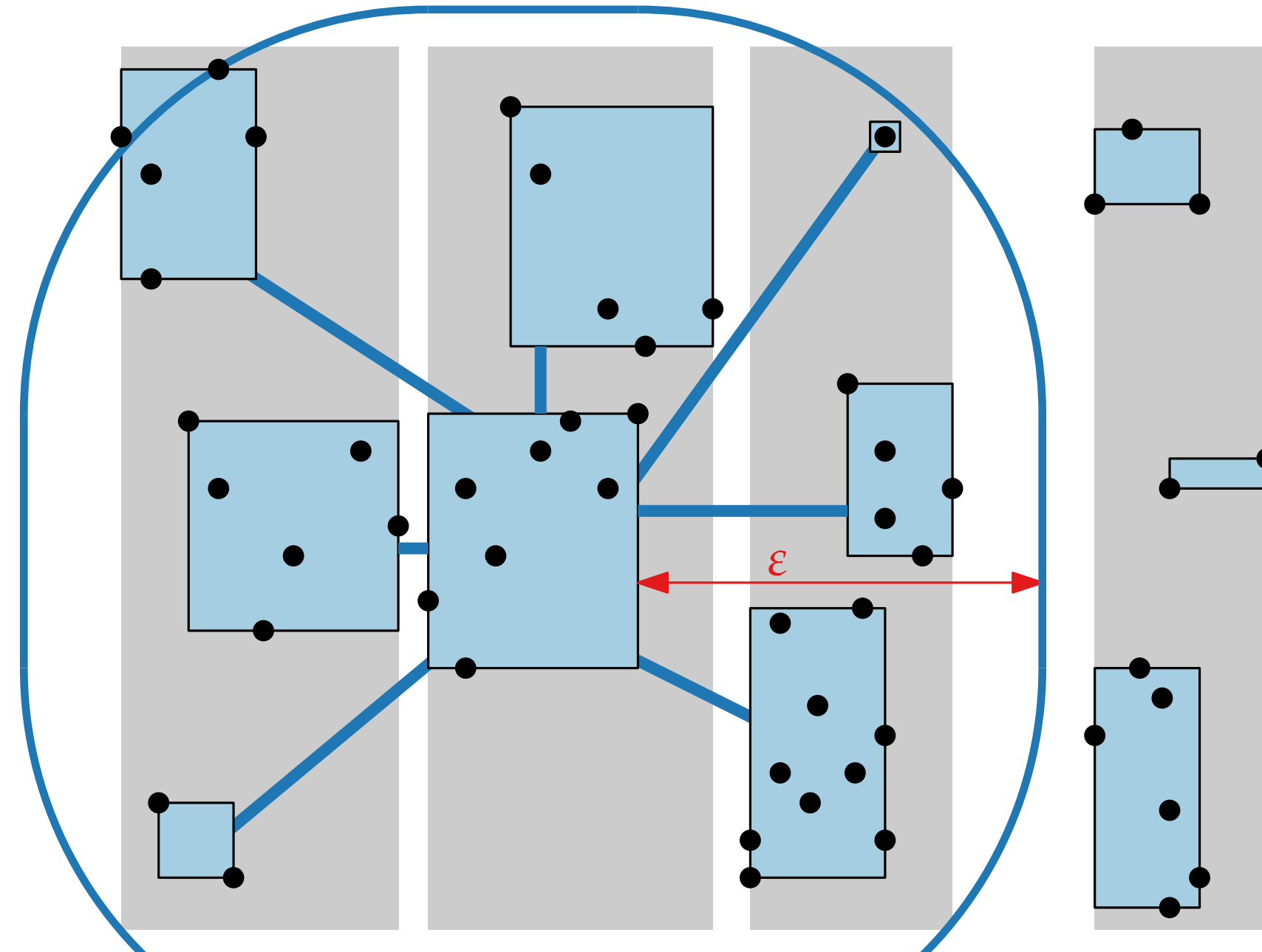
- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben?

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \longleftrightarrow \\ \varepsilon/\sqrt{2}: \longleftrightarrow \end{array}$$

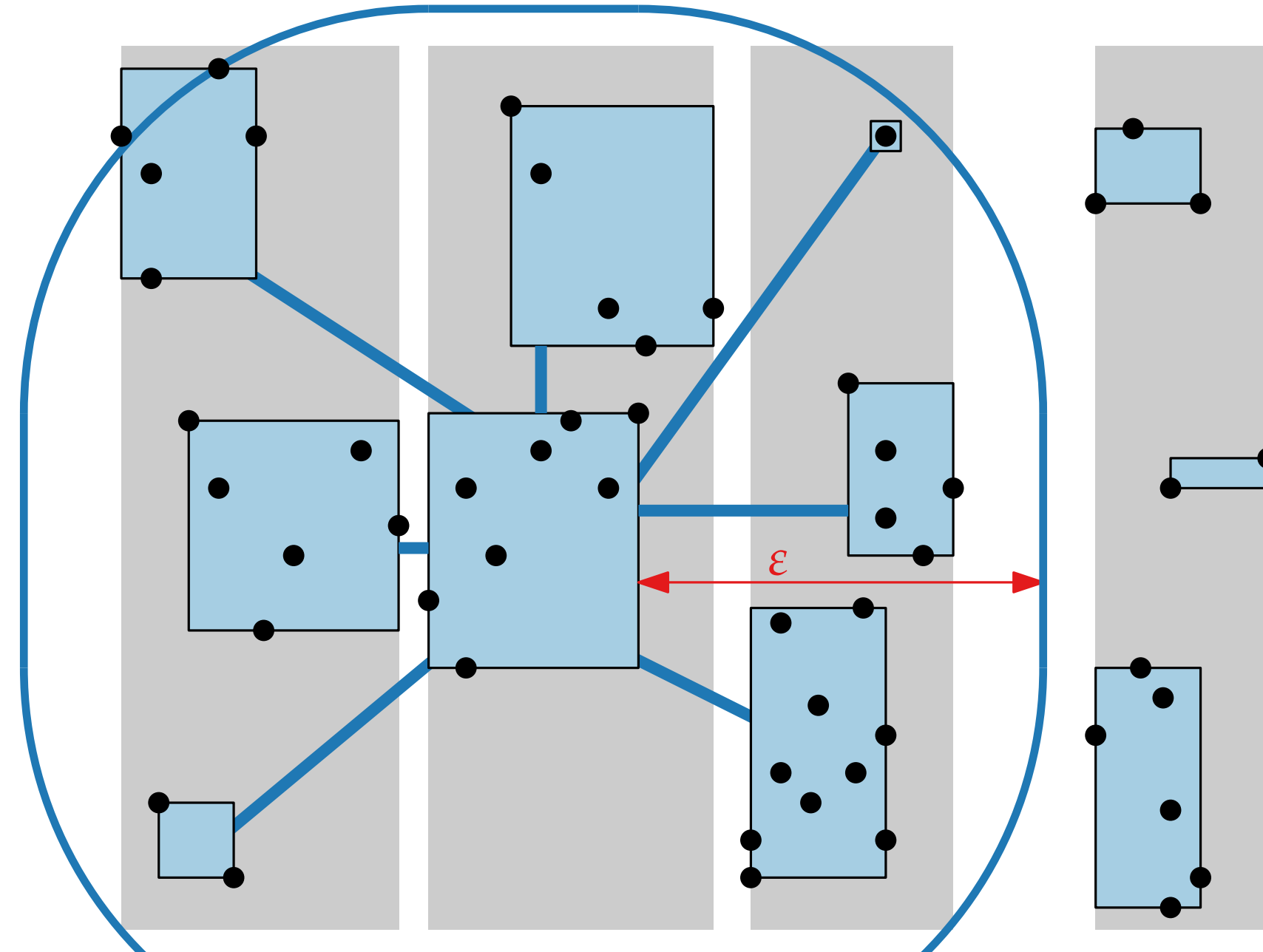


Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? $\in \mathcal{O}(1)$



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$

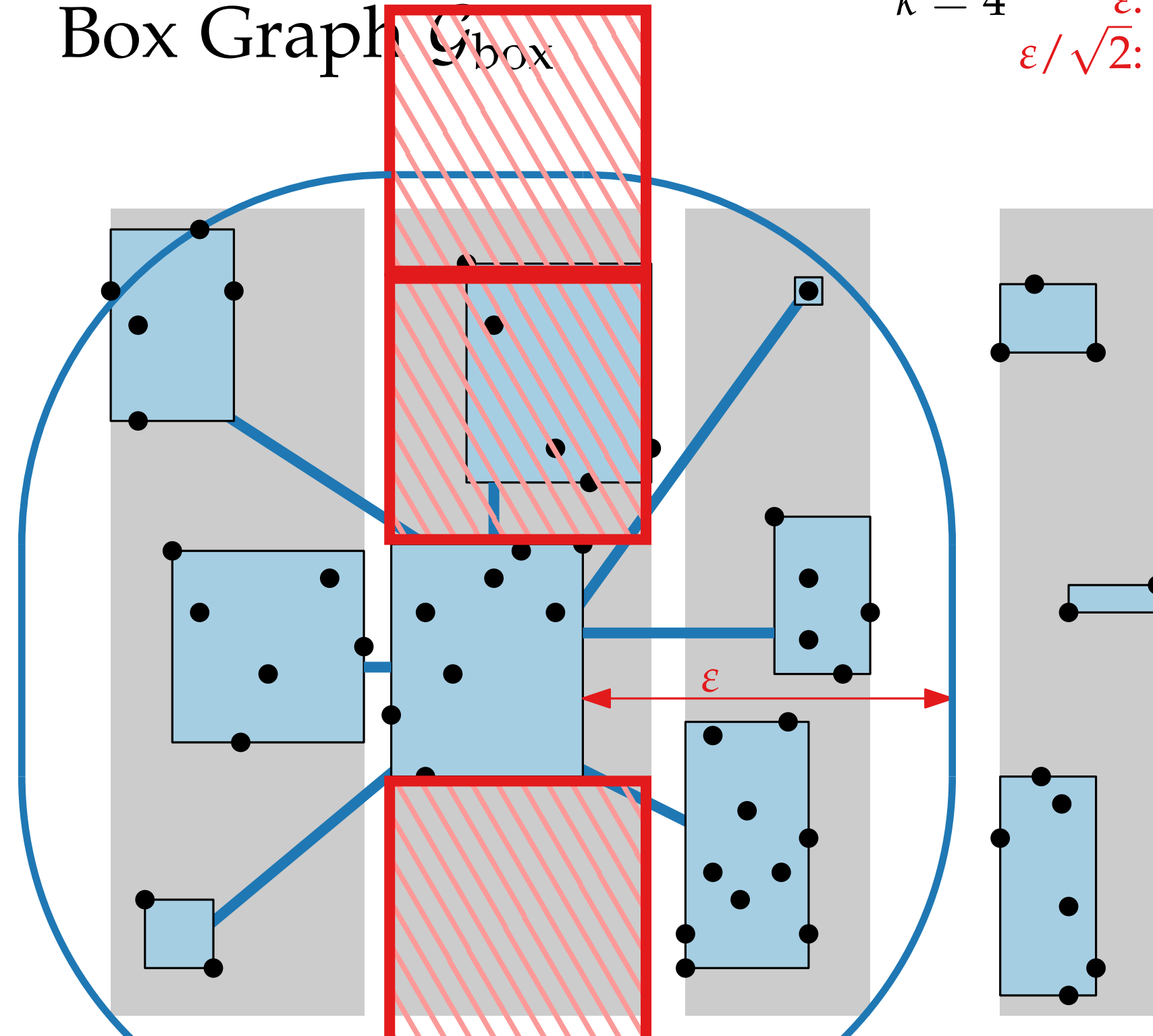


Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? $\in \mathcal{O}(1)$



Box Graph $\mathcal{G}_{\text{box}}$

$k = 4$

ε : 
 $\varepsilon/\sqrt{2}$: 

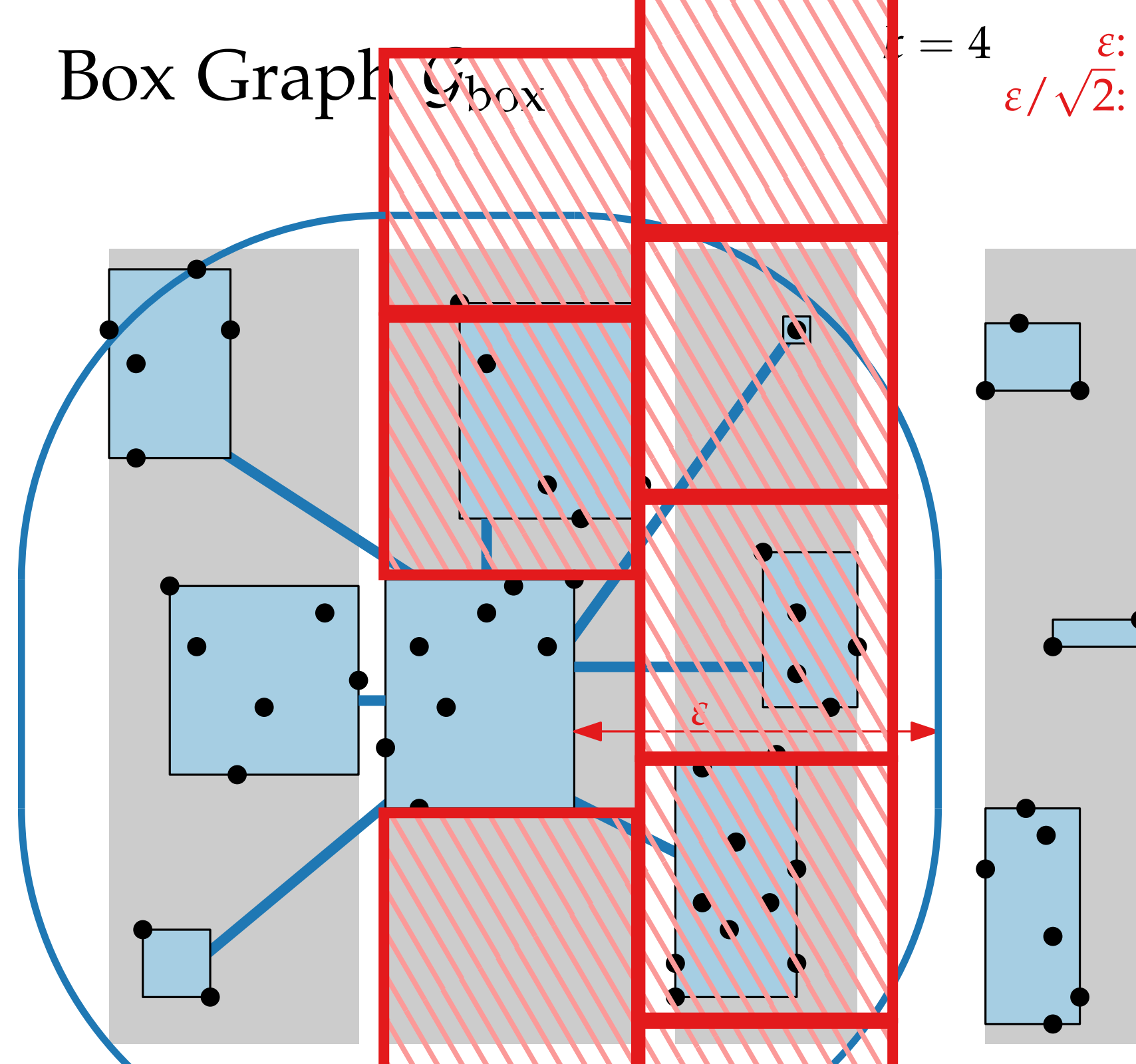


Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

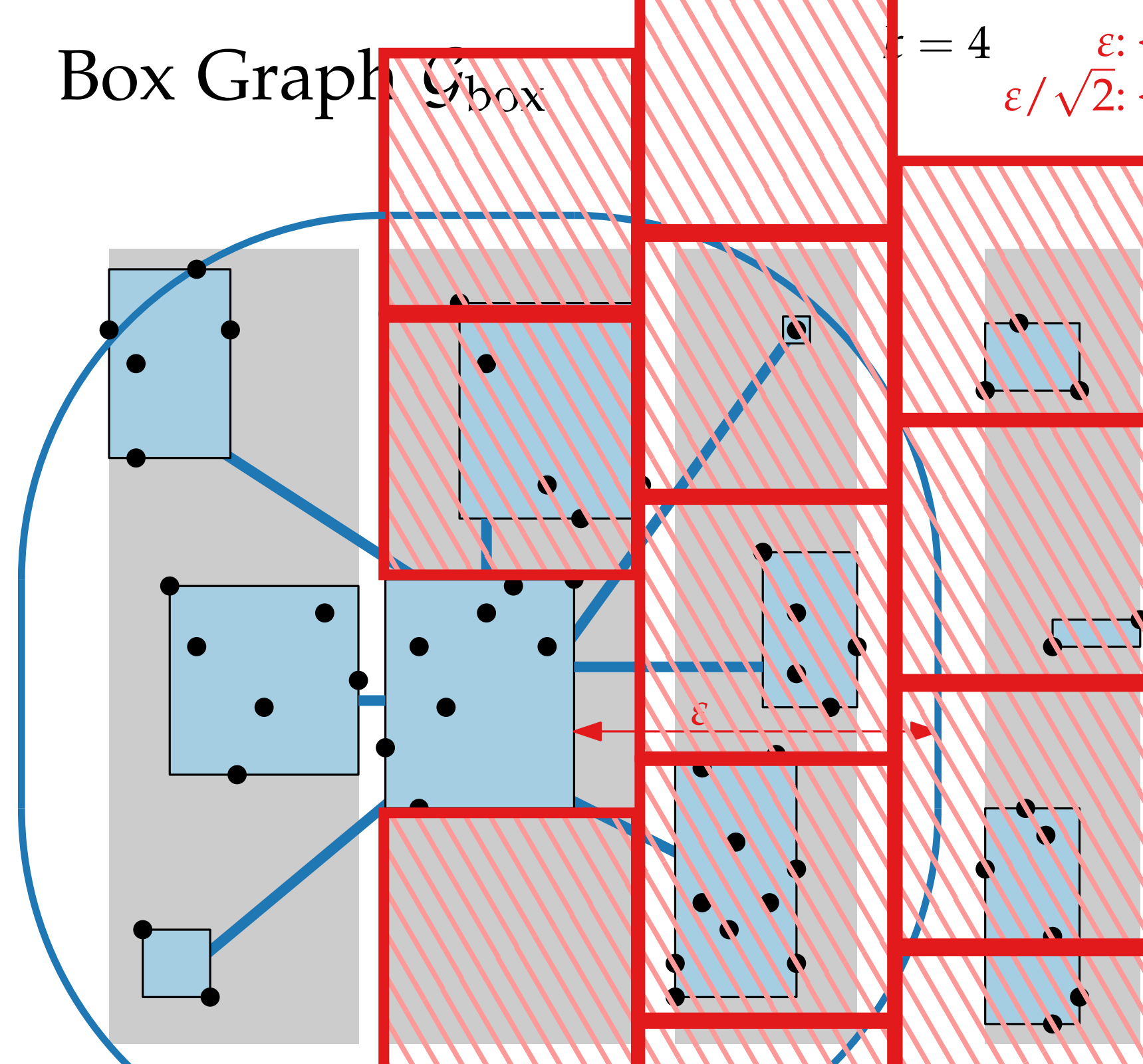
- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? $\in \mathcal{O}(1)$



Box Graph $\mathcal{G}_{\text{box}}$

$k = 4$

ε : 
 $\varepsilon/\sqrt{2}$: 



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? $\in \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$k = 4$

ε : 
 $\varepsilon/\sqrt{2}$: 



Eigenschaften der Boxen.

- Alle Punkte in einer Box...
sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten...
sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten...
kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit
Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$:
Keine der Punkte liegen
in ε -Nachbarschaft.
- Wie viele Nachbarn kann
eine Box haben? $\in \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$k = 4$

ε : 
 $\varepsilon/\sqrt{2}$: 



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

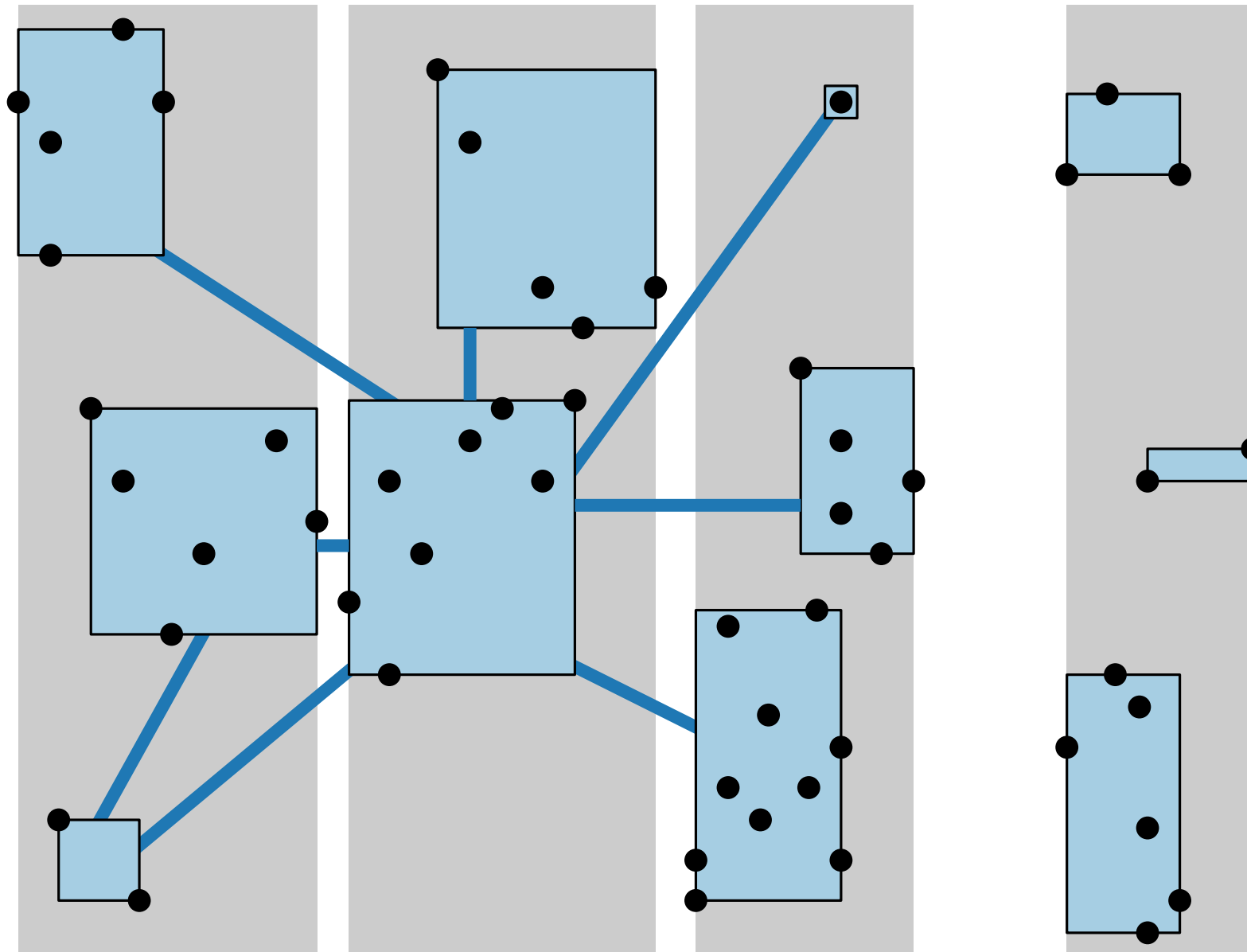
Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? **22** $\in \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \longleftrightarrow \\ \varepsilon/\sqrt{2}: \longleftrightarrow \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

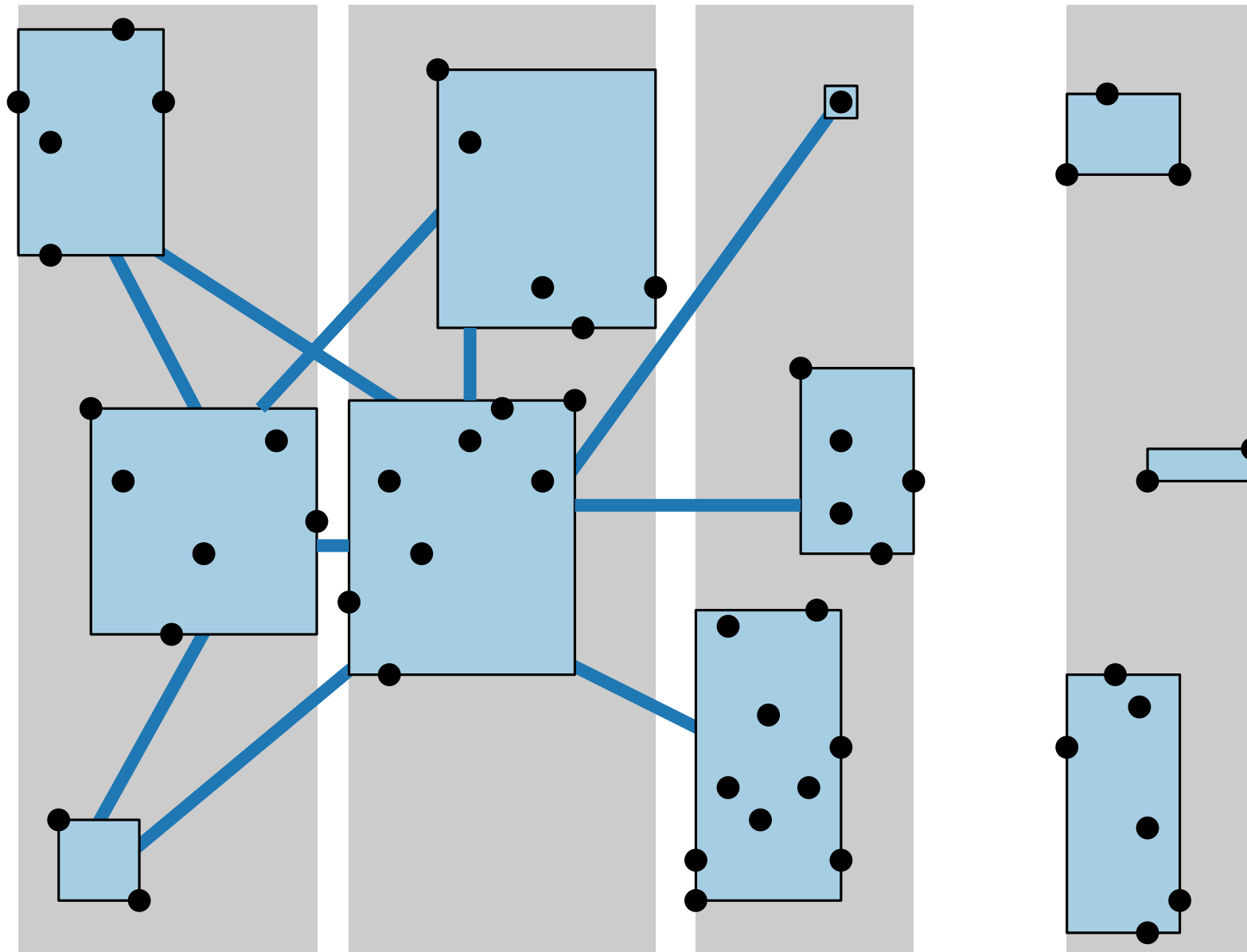
Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? **22** $\in \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \\ \varepsilon/\sqrt{2}: \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

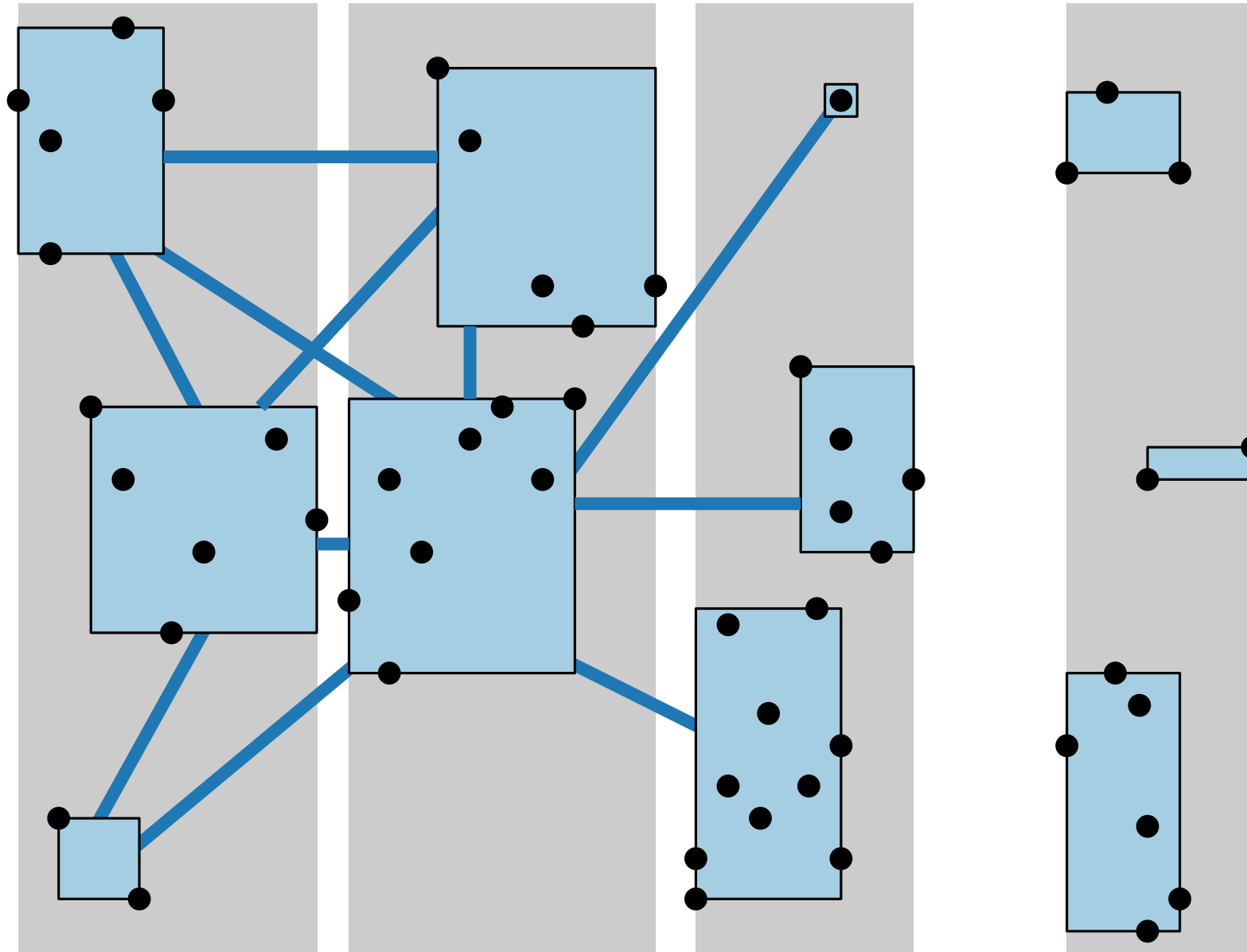
Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? **22** $\in \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\begin{array}{l} \varepsilon: \longleftrightarrow \\ \varepsilon/\sqrt{2}: \longleftrightarrow \end{array}$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

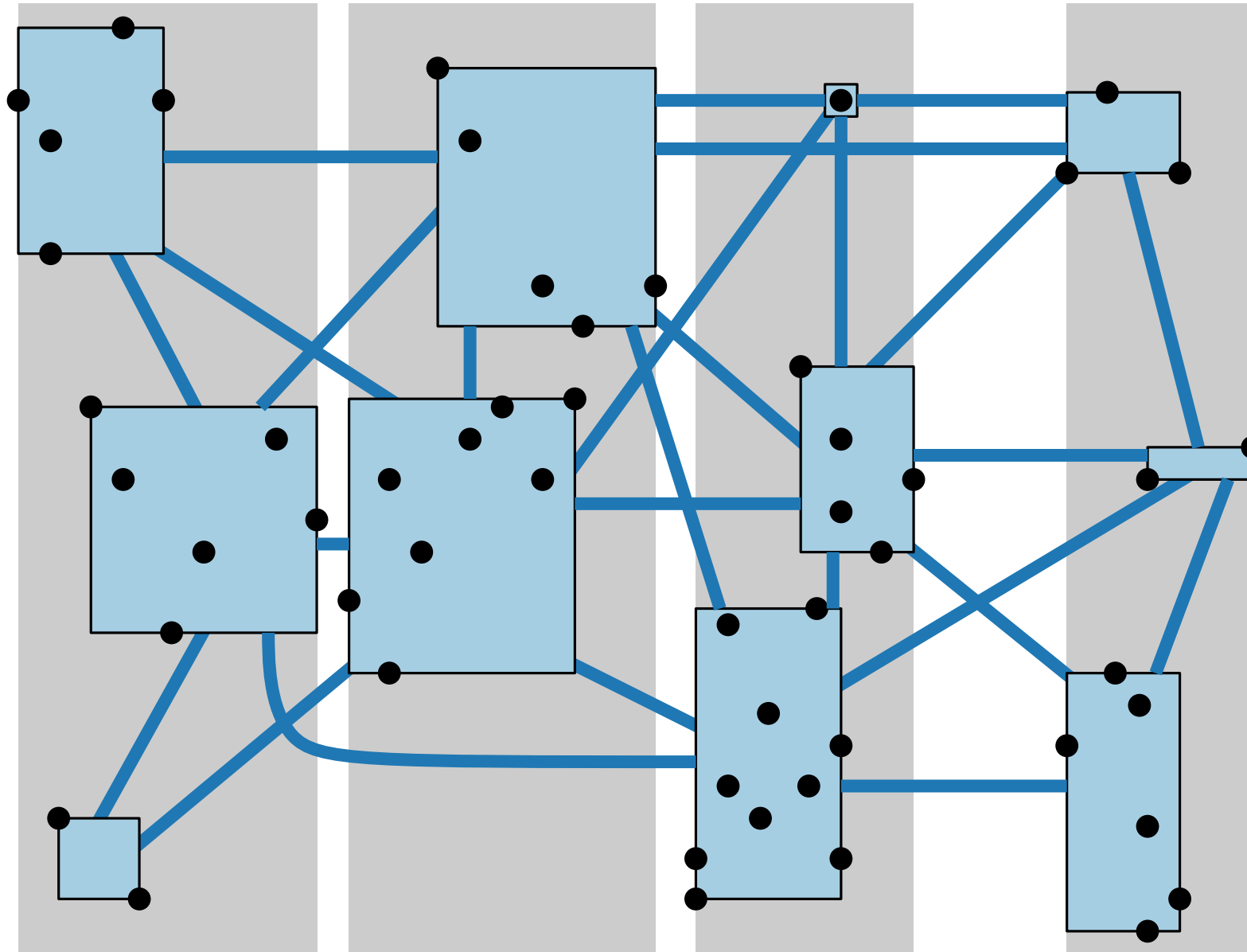
Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? **22** $\in \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Eigenschaften der Boxen.

- Alle Punkte in einer Box... sind in ε -Nachbarschaft.
(Boxbreite und -höhe $\leq \varepsilon/\sqrt{2}$)
- In Boxen mit min. k Punkten... sind alle Punkte Kerne.
- In Boxen mit $< k$ Punkten... kann es auch Kerne geben!

Eigenschaften von Boxpaaren.

- Verbinde Boxen mit Distanz $\leq \varepsilon$.
- Nicht-Nachbarn in $\mathcal{G}_{\text{box}}$: Keine der Punkte liegen in ε -Nachbarschaft.
- Wie viele Nachbarn kann eine Box haben? **22** $\in \mathcal{O}(1)$

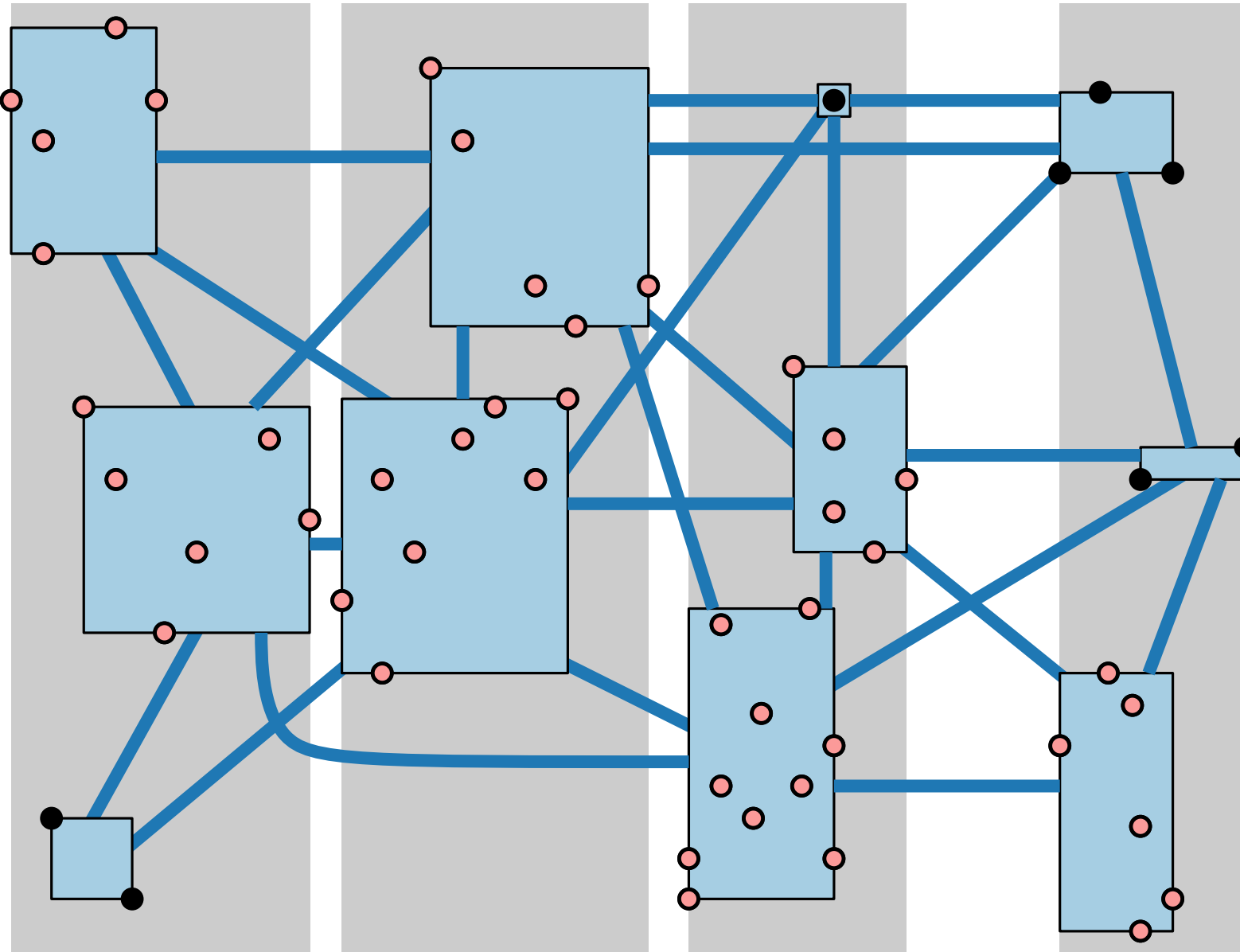
Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne



Box Graph $\mathcal{G}_{\text{box}}$

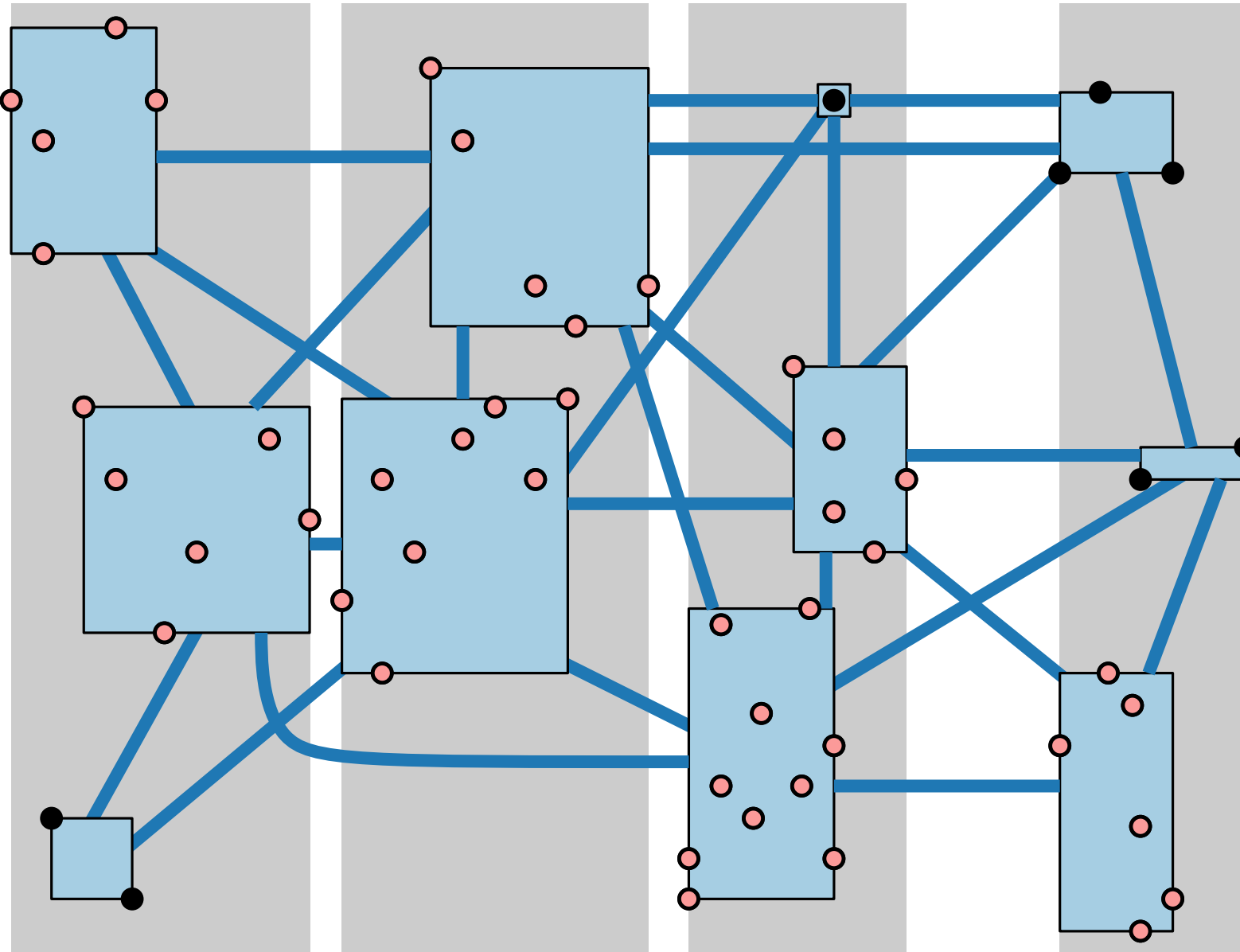
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

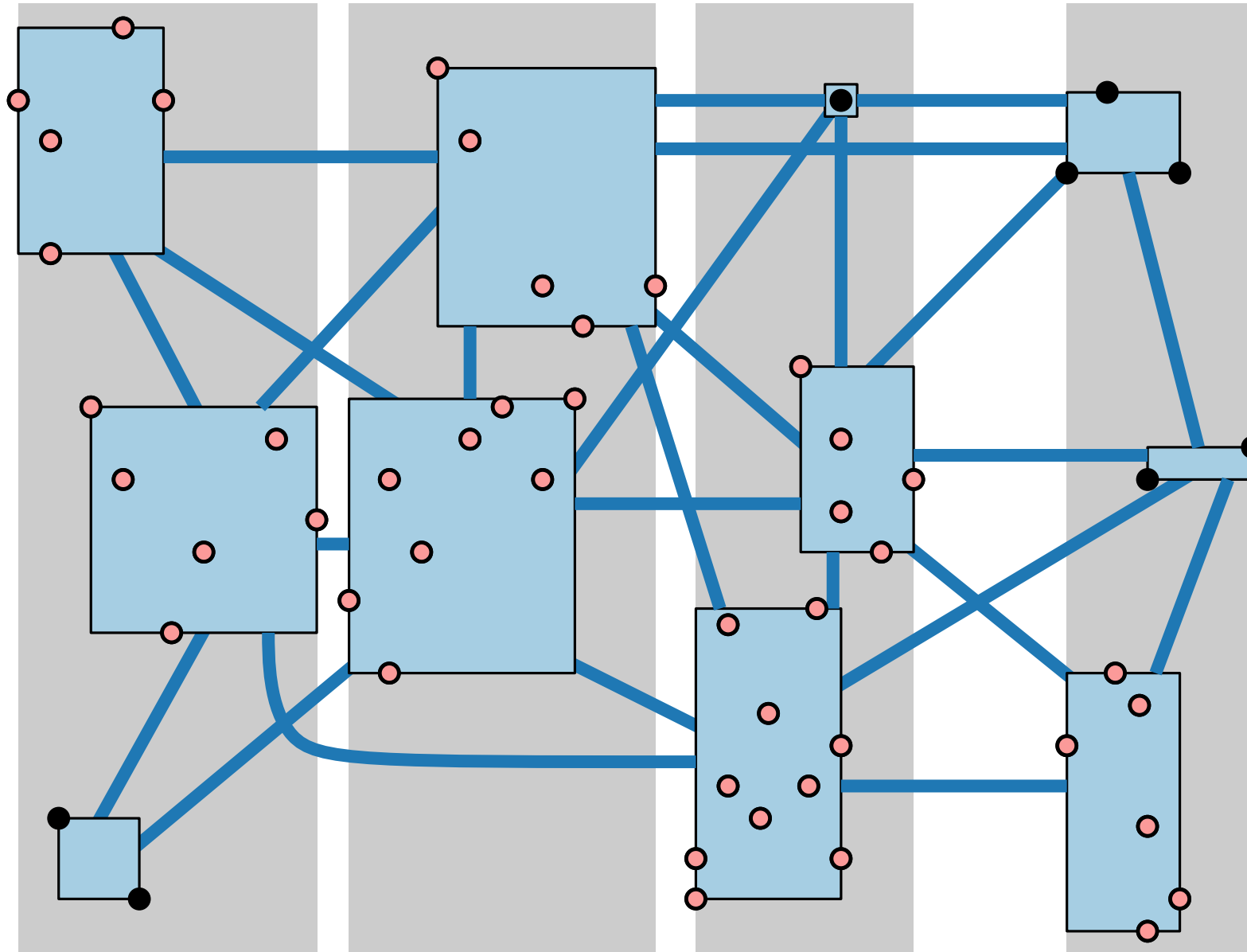
- Haben bereits alle Kerne in „schweren“ Boxen



Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



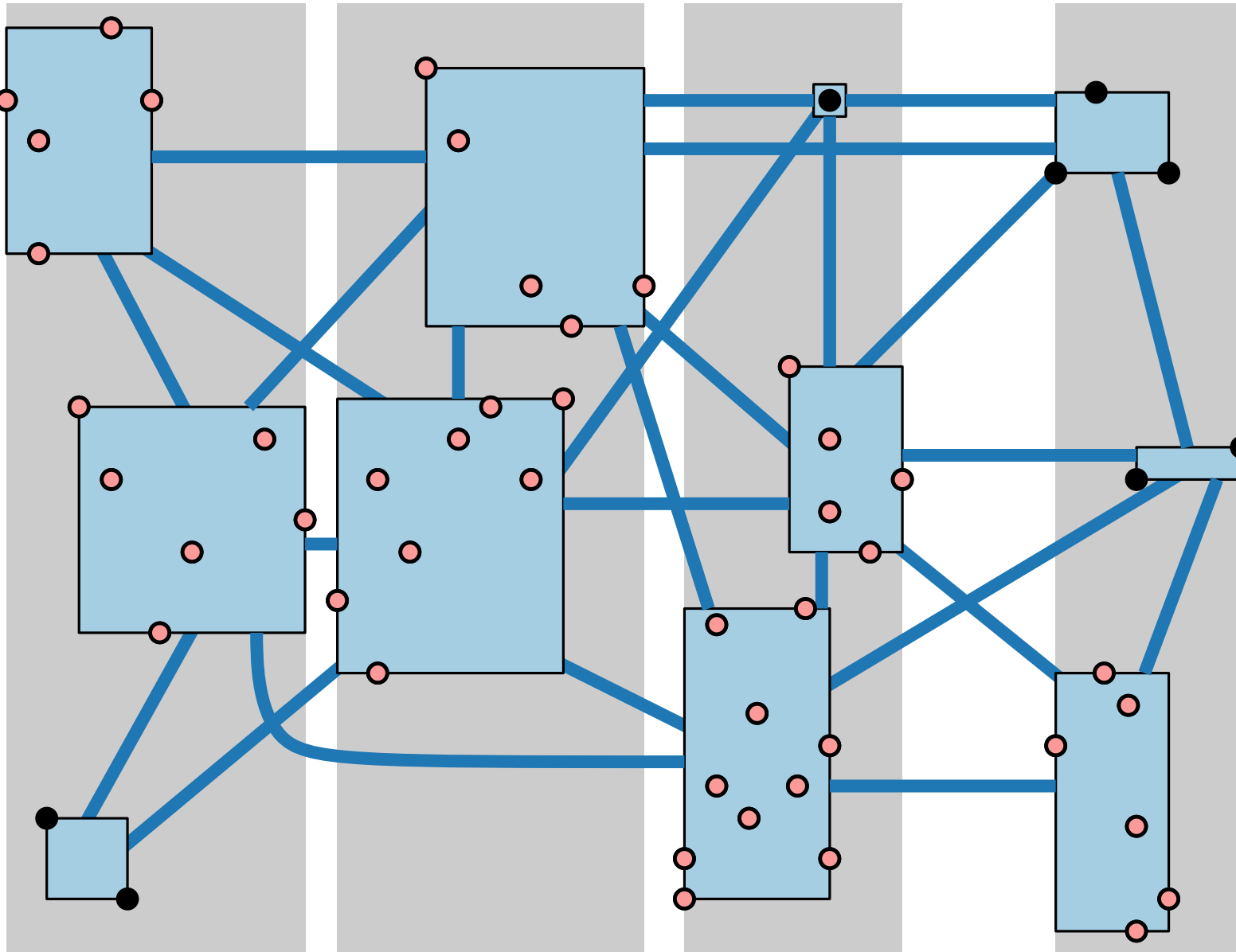
2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



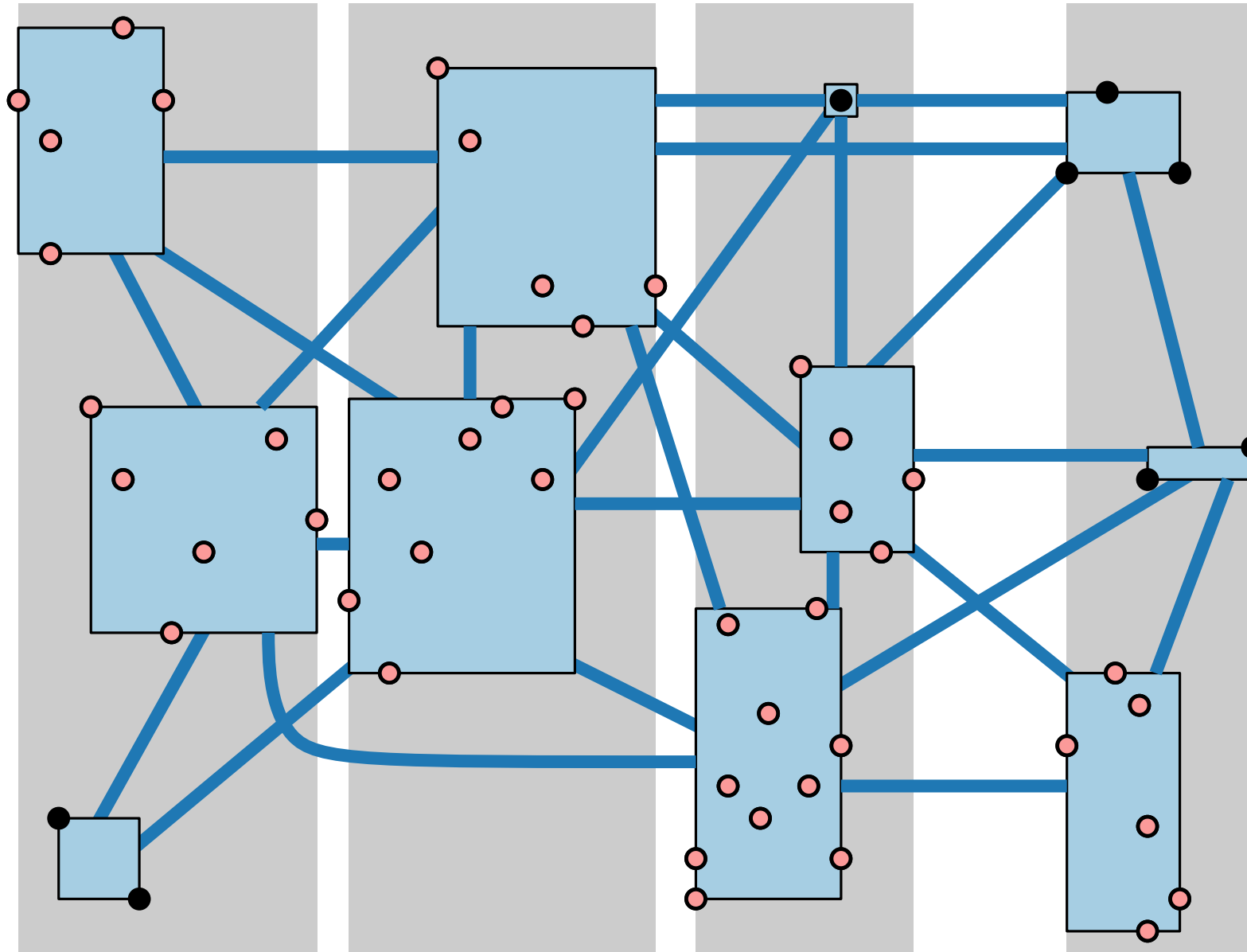
2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



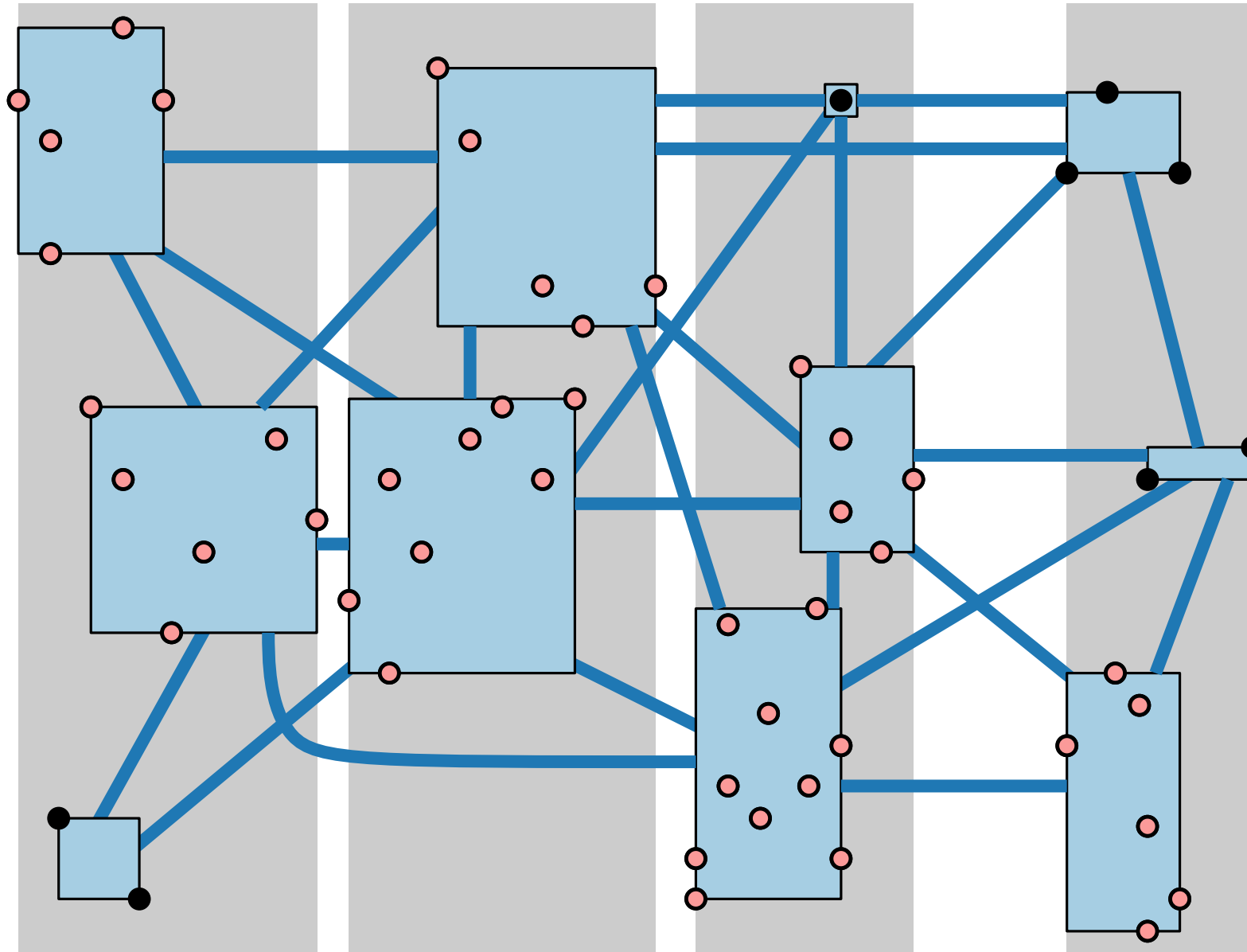
2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

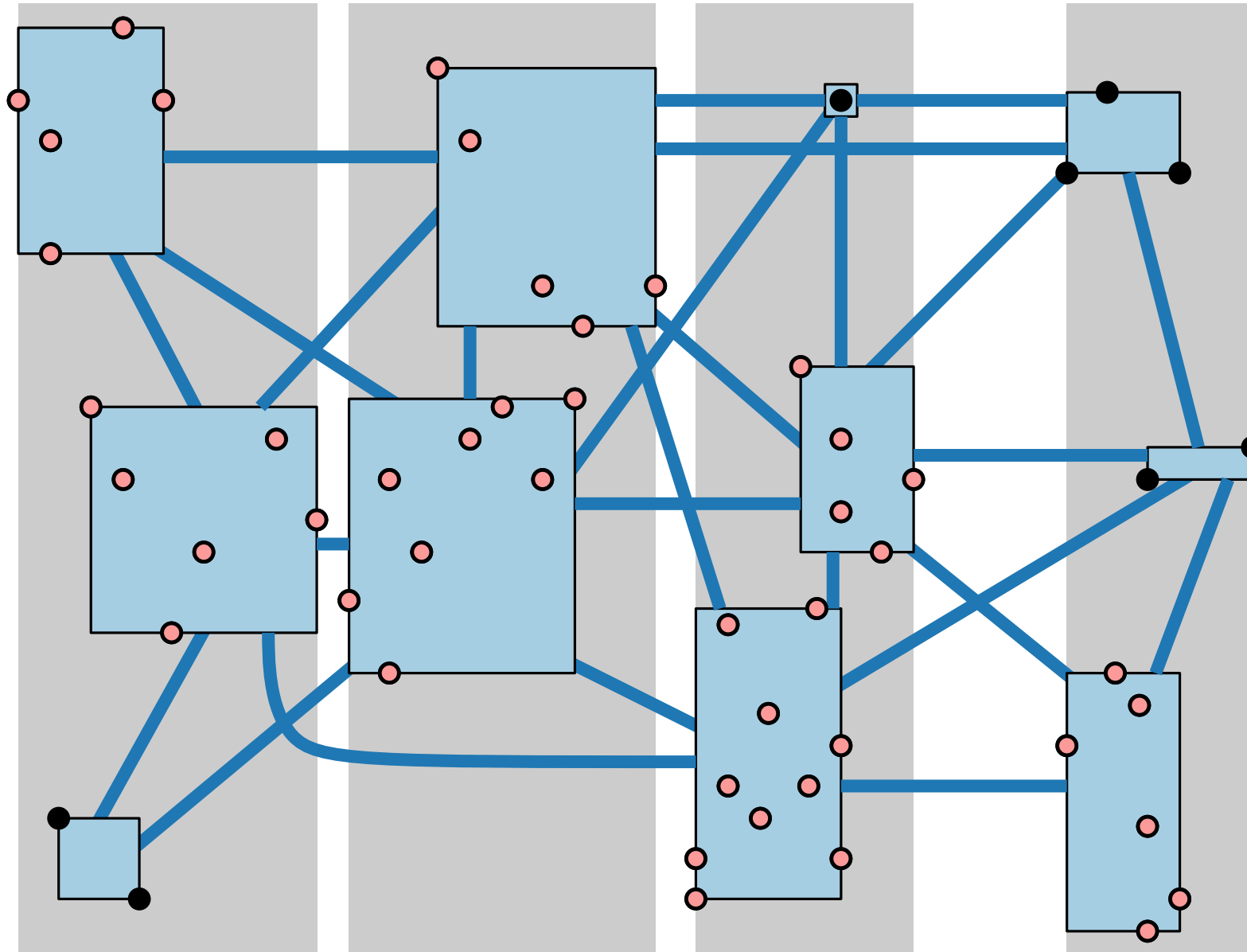
- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
 - ... teste alle Paare.

Laufzeit?

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

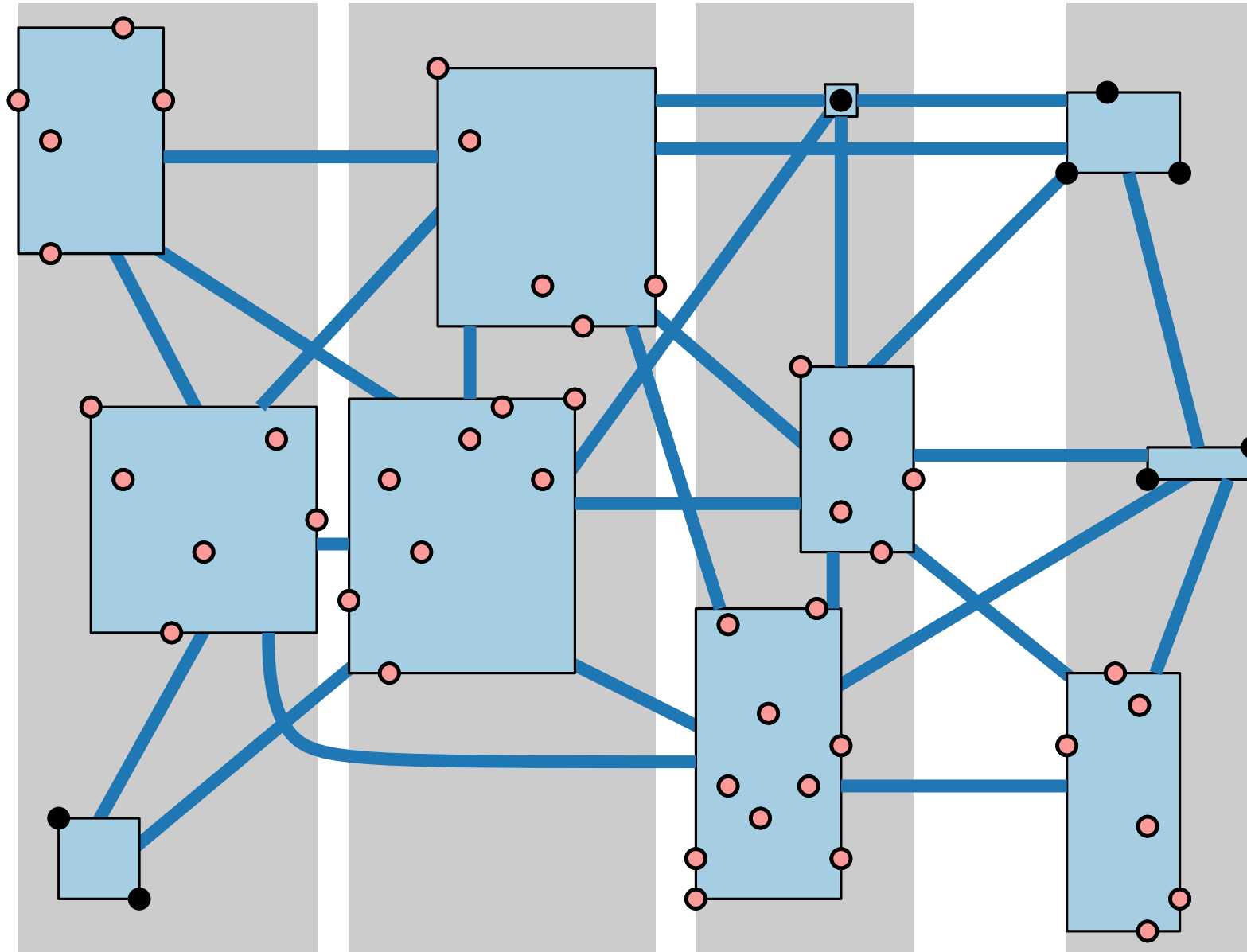
Laufzeit?

- Andere Box ist auch leicht:

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

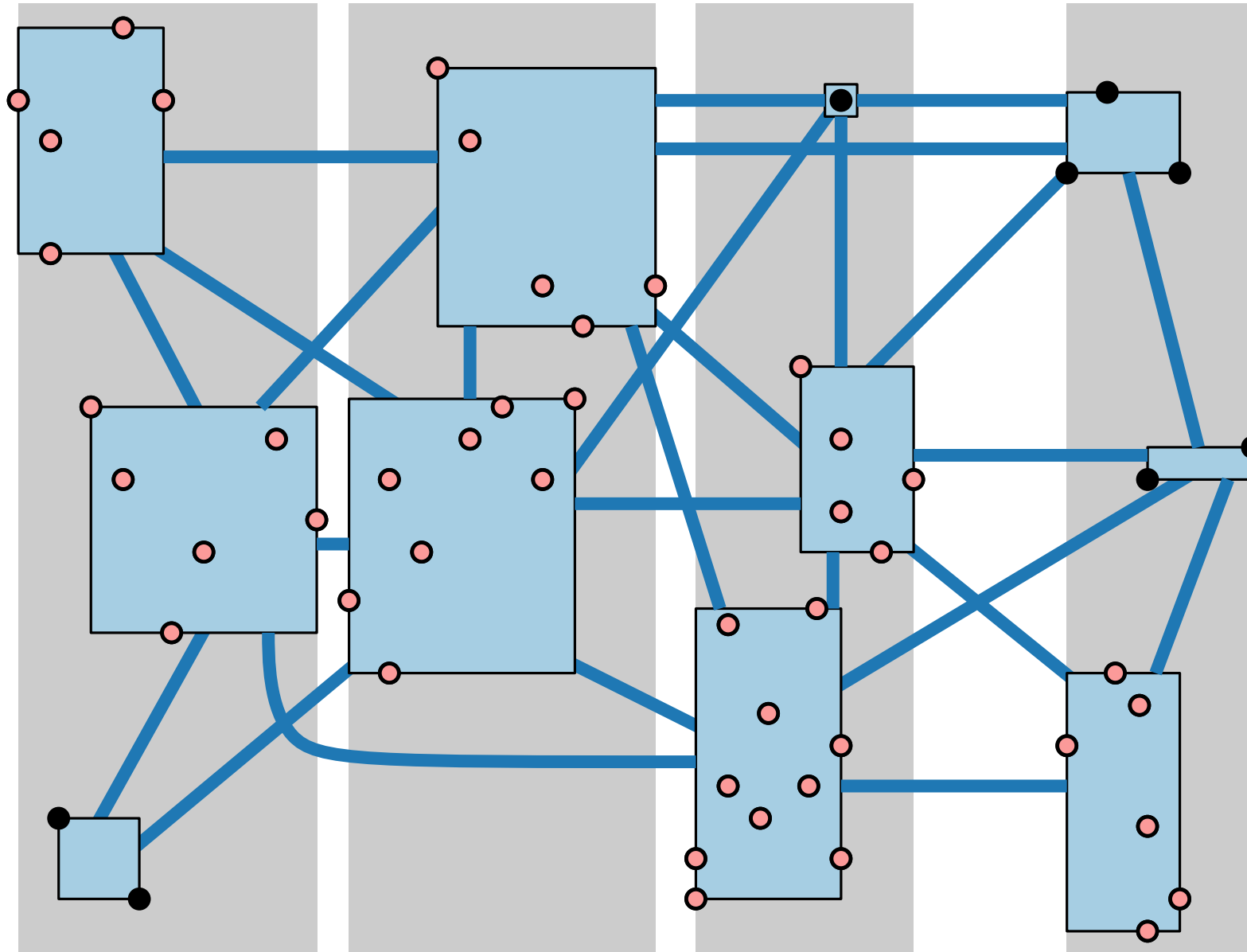
Laufzeit?

- Andere Box ist auch leicht:
 $\mathcal{O}(k^2) = \mathcal{O}(1)$

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

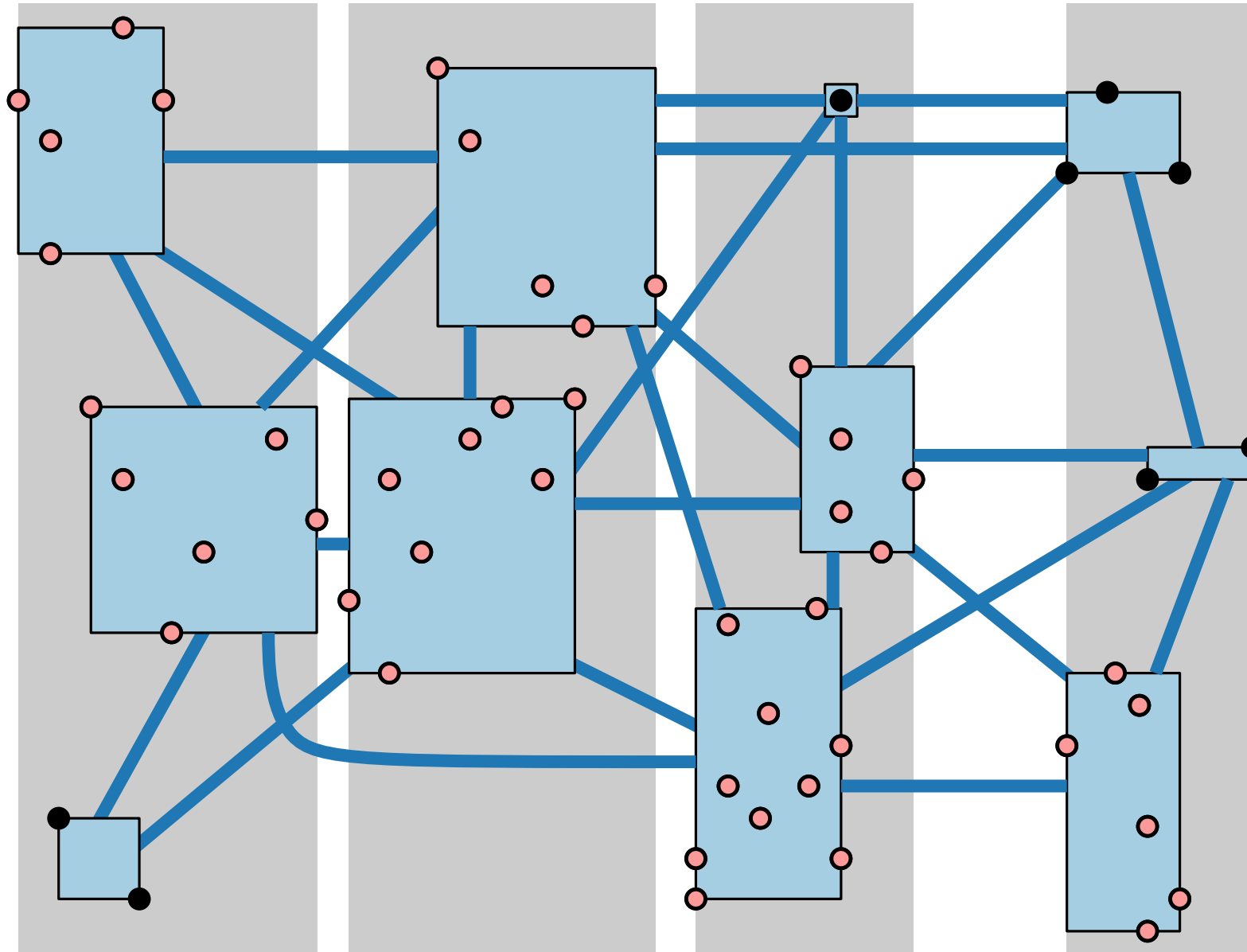
Laufzeit?

- Andere Box ist auch leicht:
 $\mathcal{O}(k^2) = \mathcal{O}(1)$
- Andere Box ist schwer:

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

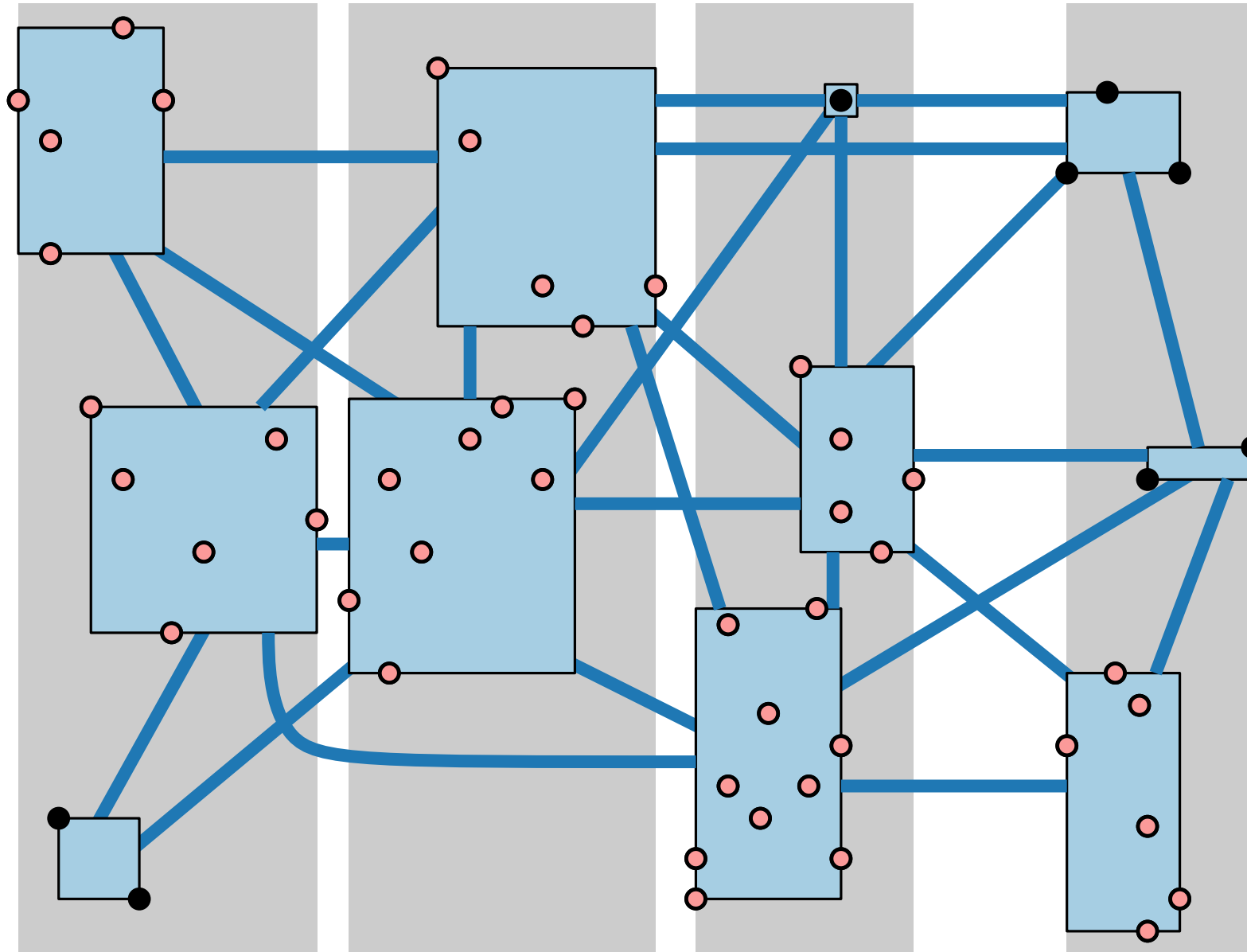
Laufzeit?

- Andere Box ist auch leicht:
 $\mathcal{O}(k^2) = \mathcal{O}(1)$
- Andere Box ist schwer:
Lasse schwere Box bezahlen:

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

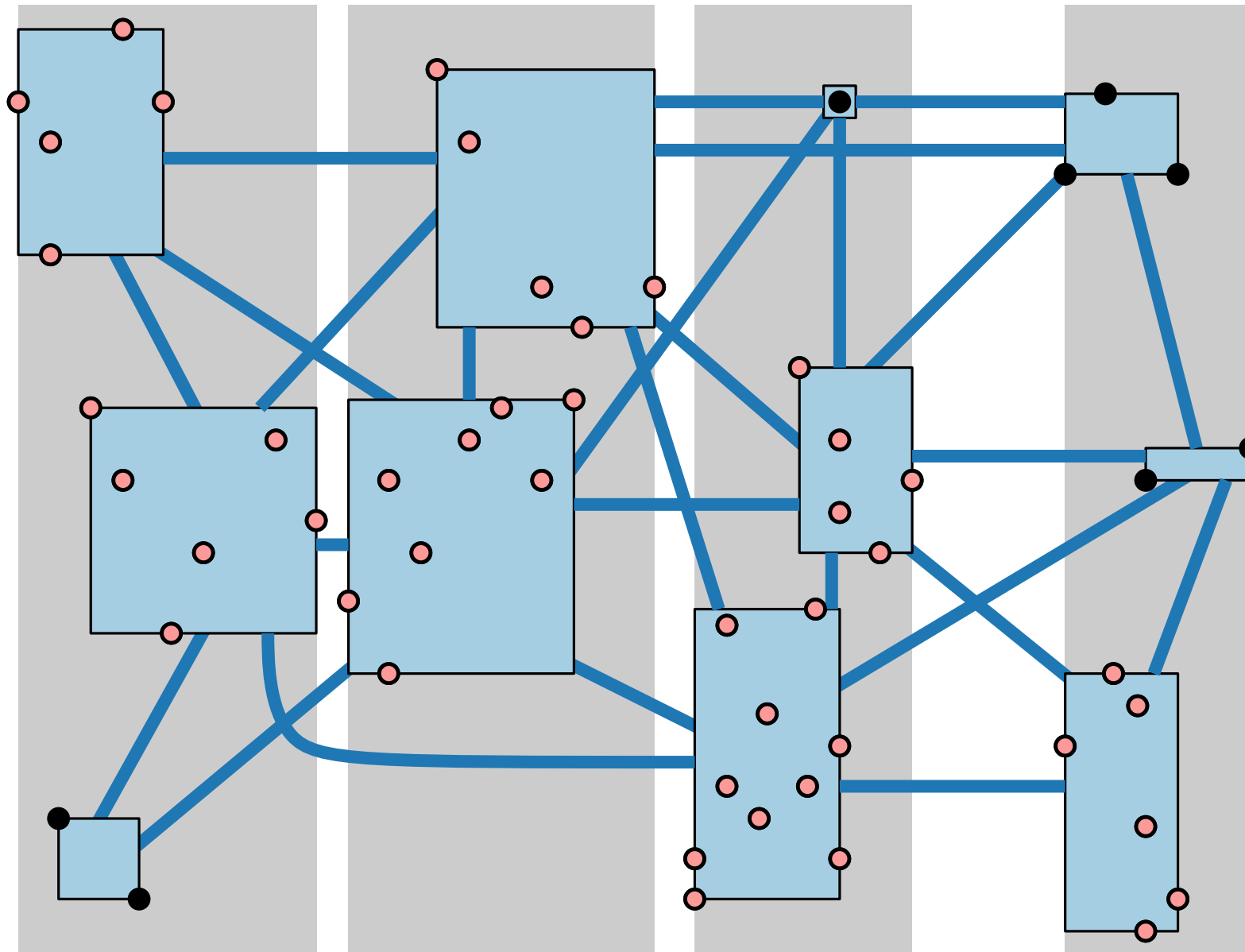
Laufzeit?

- Andere Box ist auch leicht:
 $\mathcal{O}(k^2) = \mathcal{O}(1)$
- Andere Box ist schwer:
Lasse schwere Box bezahlen:
Punkte in schweren Boxen
werden $\leq 22k$ mal überprüft
(!!)

Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



2. Finde alle Kerne

- Haben bereits alle Kerne in „schweren“ Boxen
- Für alle „leichten“ Boxen:
 - für alle Nachbarn:
... teste alle Paare.

Laufzeit?

- Andere Box ist auch leicht:
 $\mathcal{O}(k^2) = \mathcal{O}(1)$
- Andere Box ist schwer:
Lasse schwere Box bezahlen:
Punkte in schweren Boxen
werden $\leq 22k$ mal überprüft
(!!) $\rightarrow \mathcal{O}(kn) = \mathcal{O}(n)$

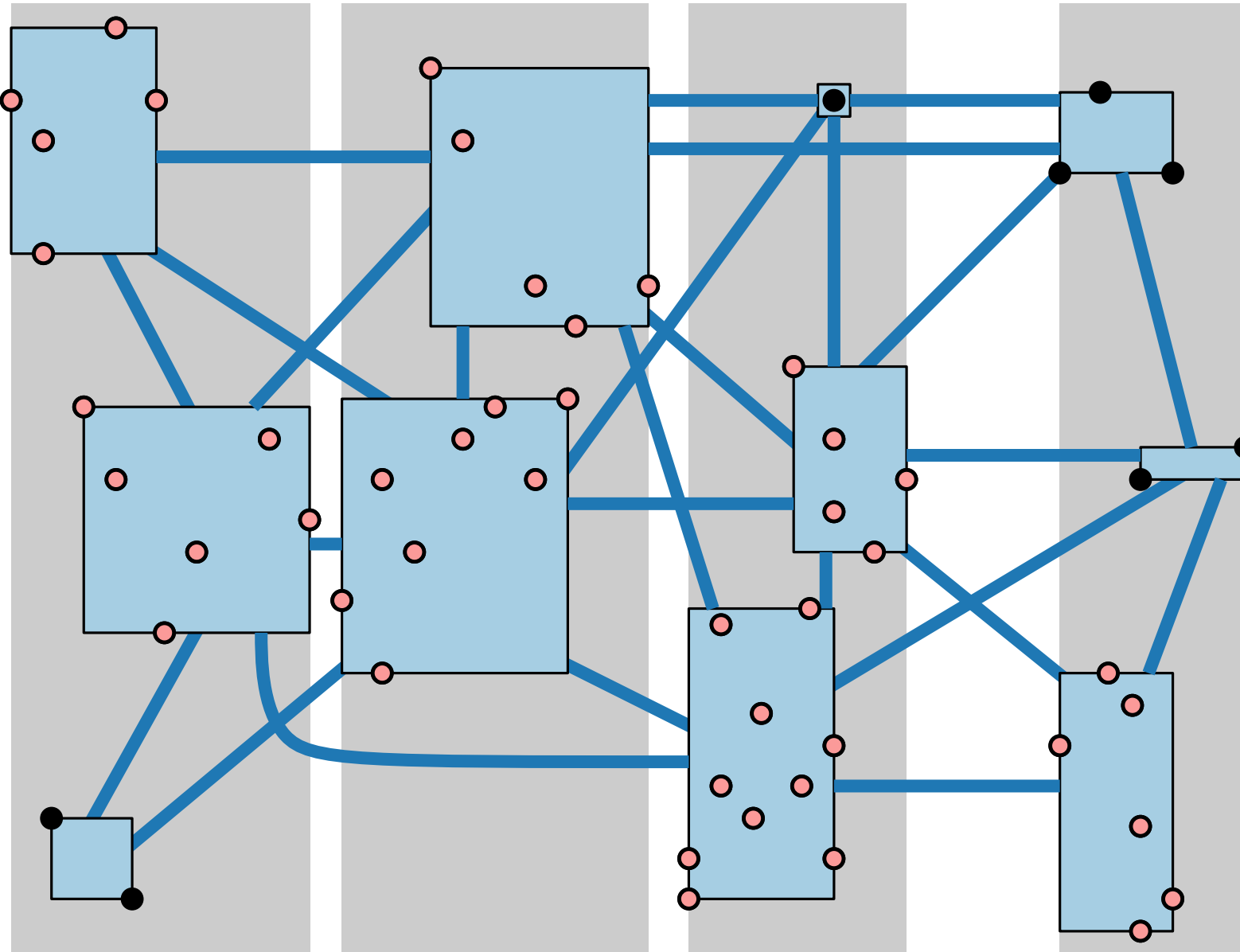
Box Graph $\mathcal{G}_{\text{box}}$

$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Paare von schweren Boxen.



Box Graph $\mathcal{G}_{\text{box}}$

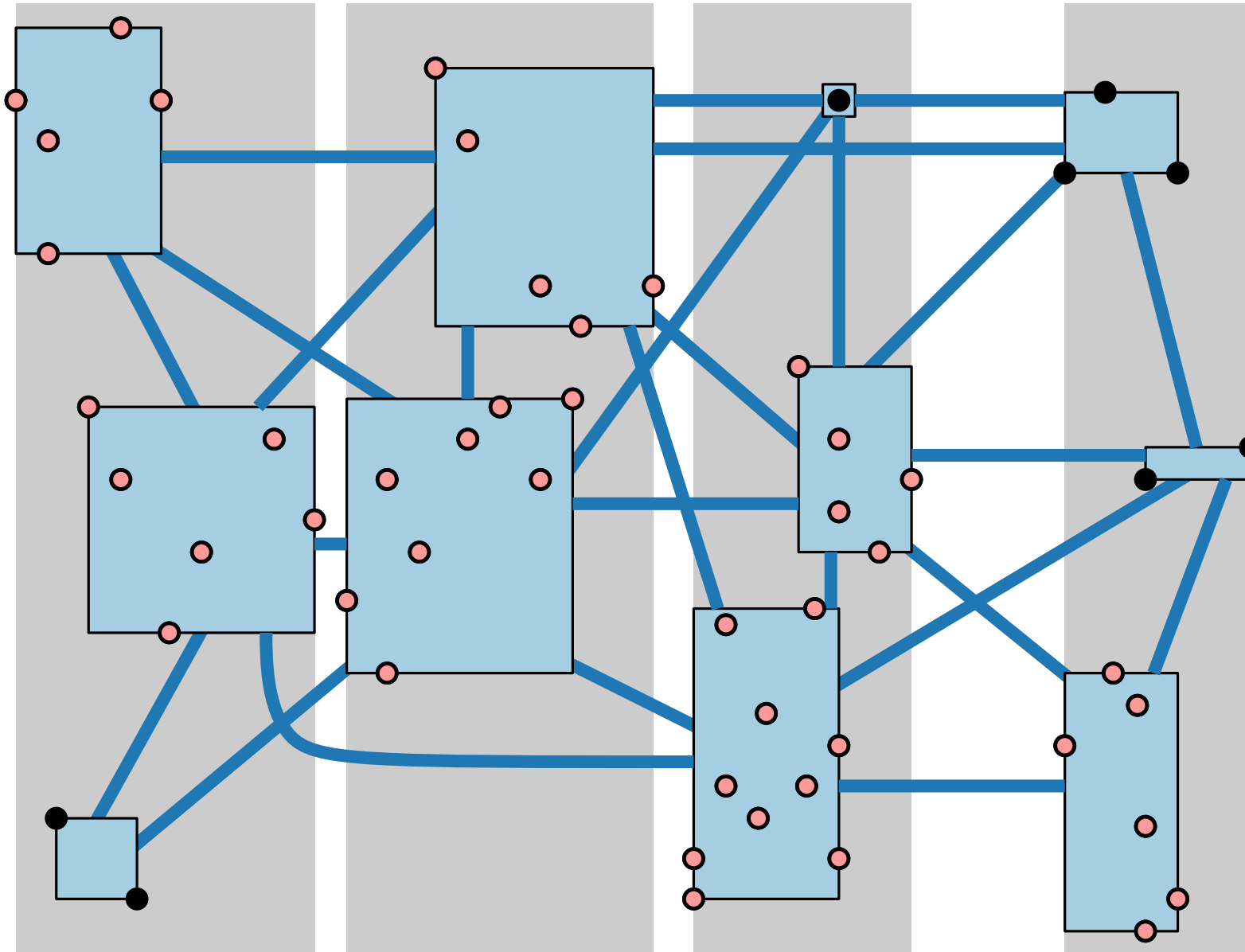
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Paare von schweren Boxen.

■ Das sind alle Kerne.



Box Graph $\mathcal{G}_{\text{box}}$

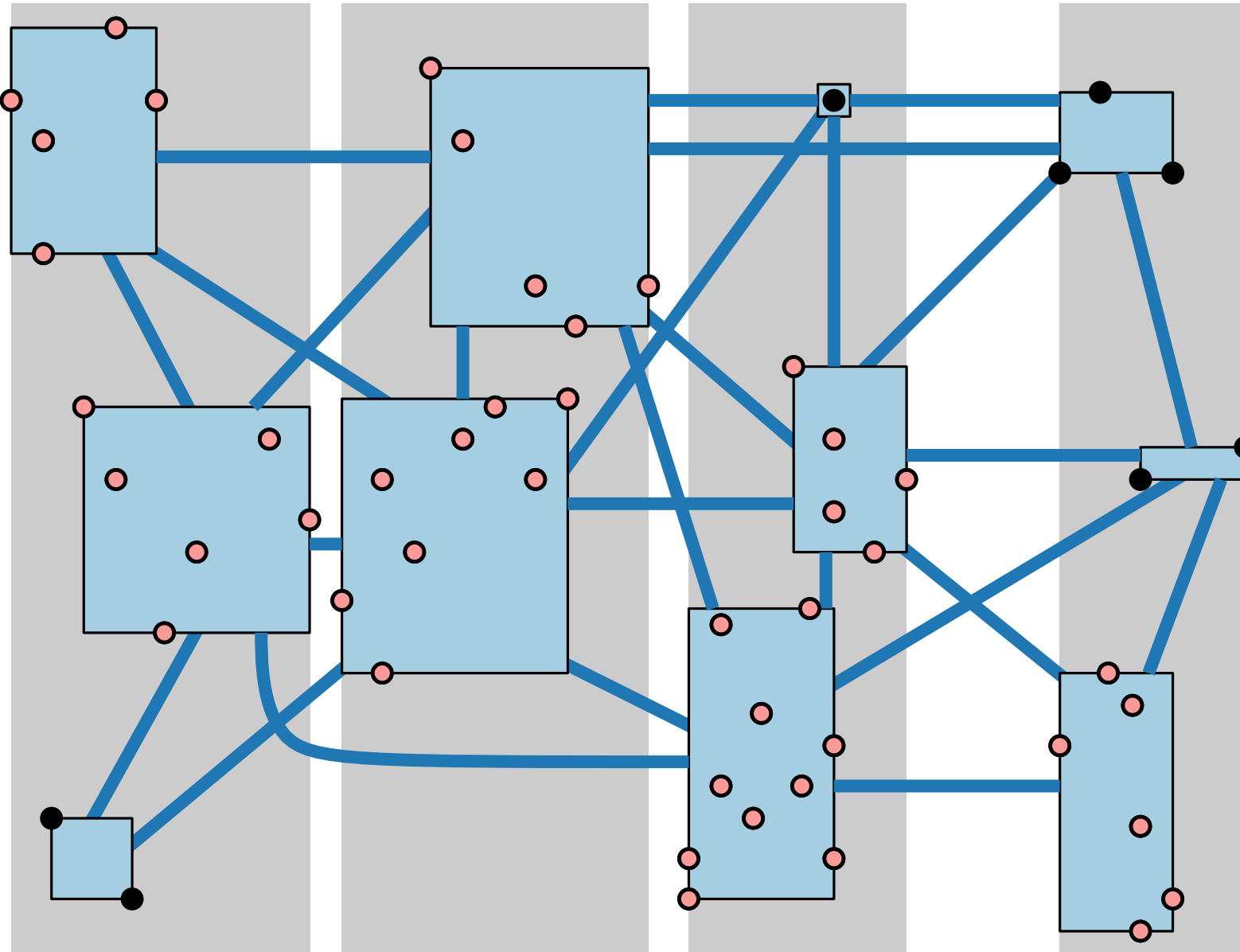
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



Box Graph $\mathcal{G}_{\text{box}}$

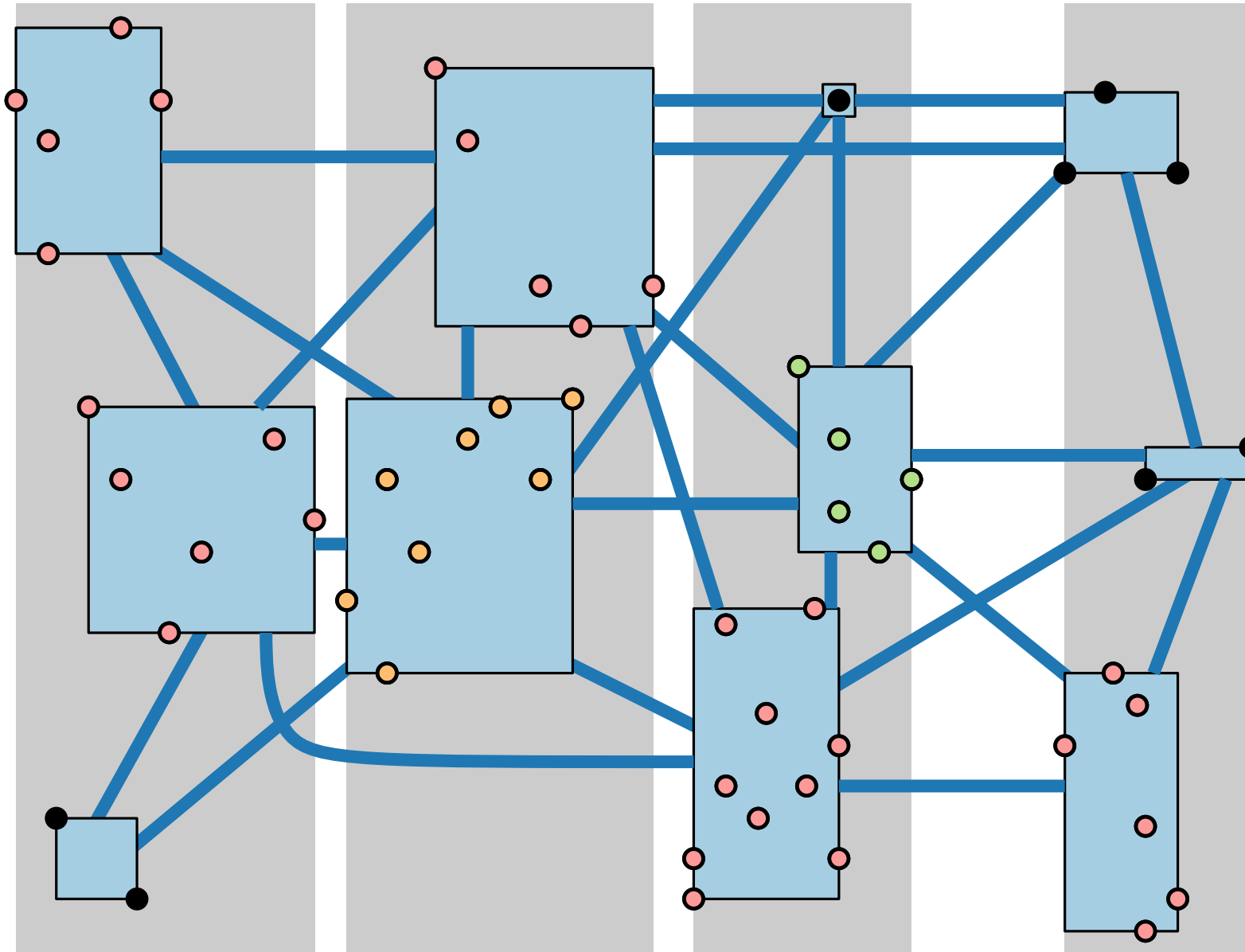
$$k = 4$$

$$\varepsilon: \longleftrightarrow$$
$$\varepsilon/\sqrt{2}: \longleftrightarrow$$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?

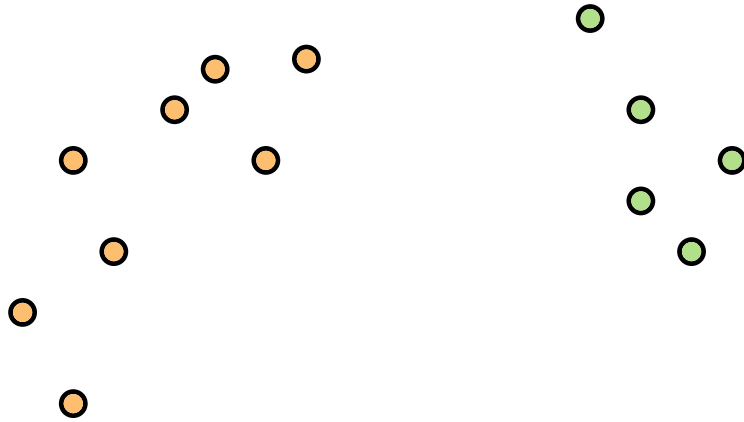


Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



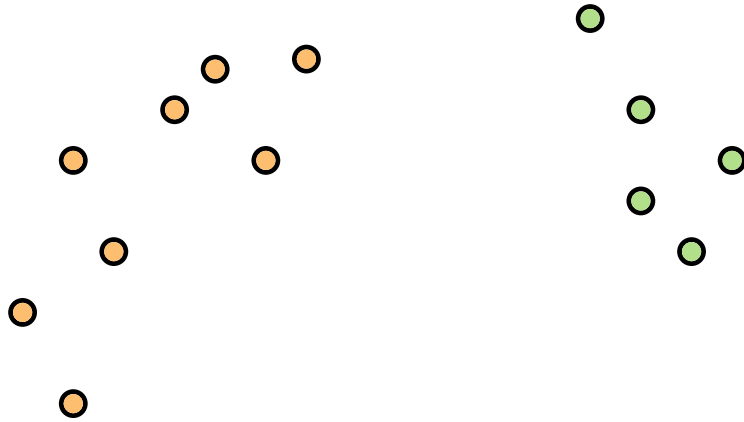
Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?

BICHROMATISCHES NÄHESTES PAAR



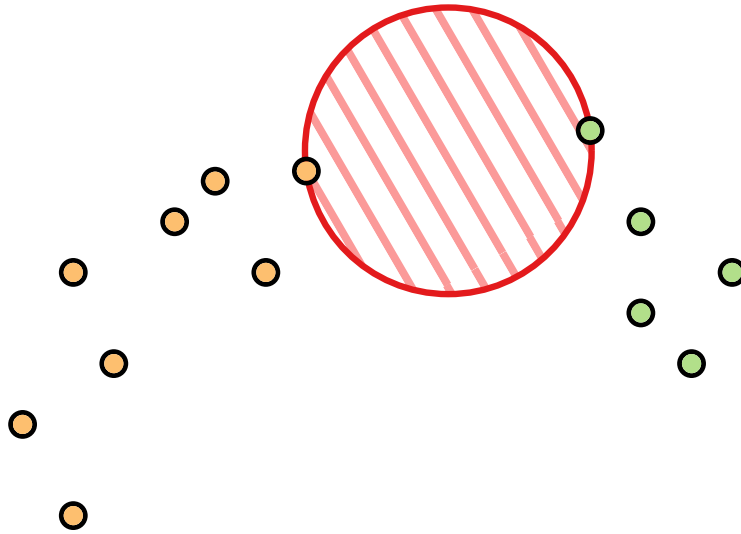
Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?

BICHROMATISCHES NÄHESTES PAAR



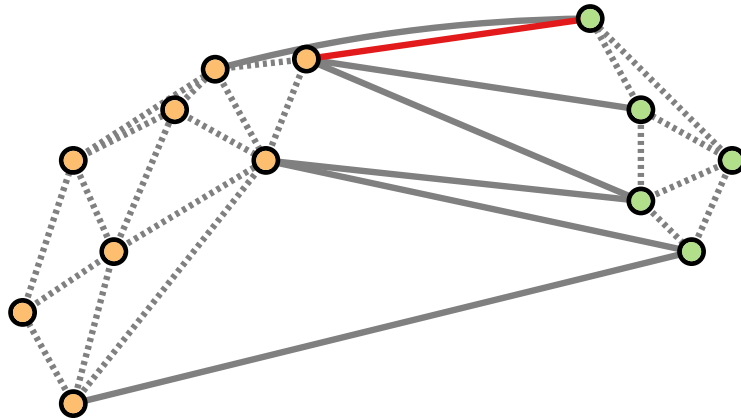
Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?

BICHROMATISCHES NÄHESTES PAAR



Box Graph $\mathcal{G}_{\text{box}}$

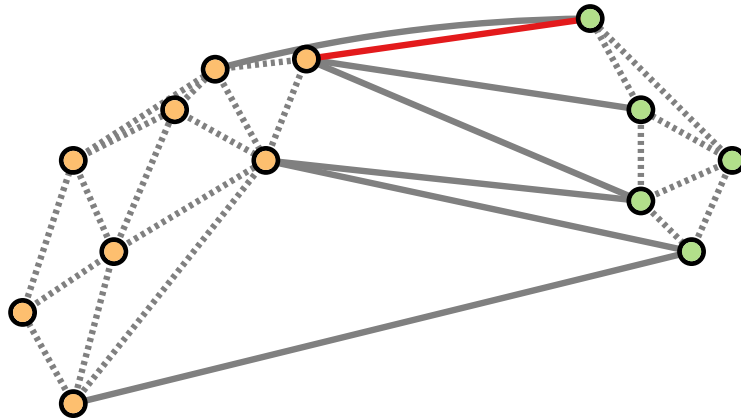


Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?

BICHROMATISCHES NÄHESTES PAAR

- Delaunay Triangulierung (DT) enthält diese Kante

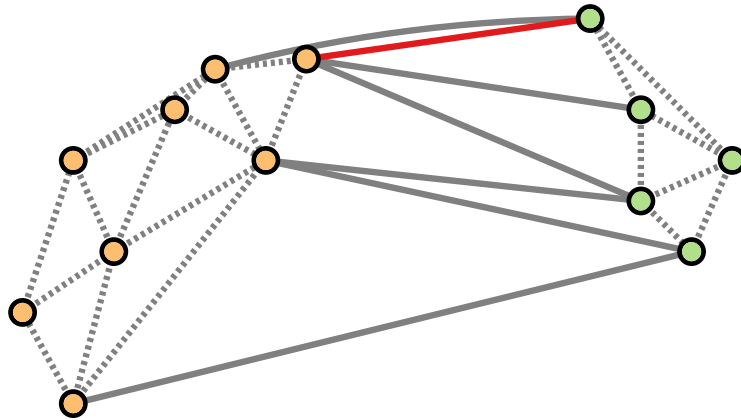


Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



BICHROMATISCHES NÄHESTES PAAR

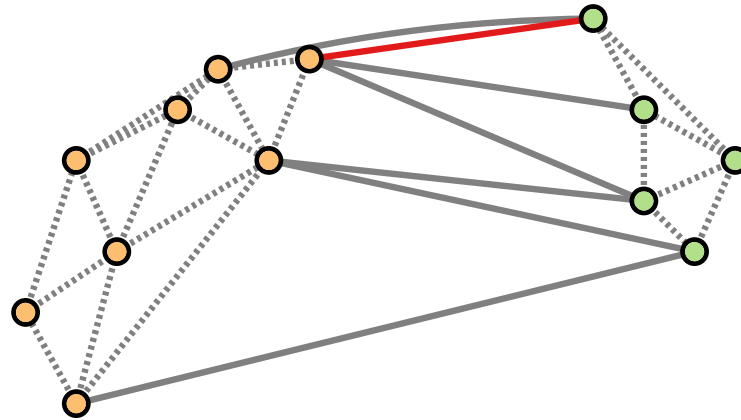
- Delaunay Triangulierung (DT) enthält diese Kante
- DT hat n Kanten, berechenbar in $\mathcal{O}(n \log n)$ Zeit

Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



BICHROMATISCHES NÄHESTES PAAR

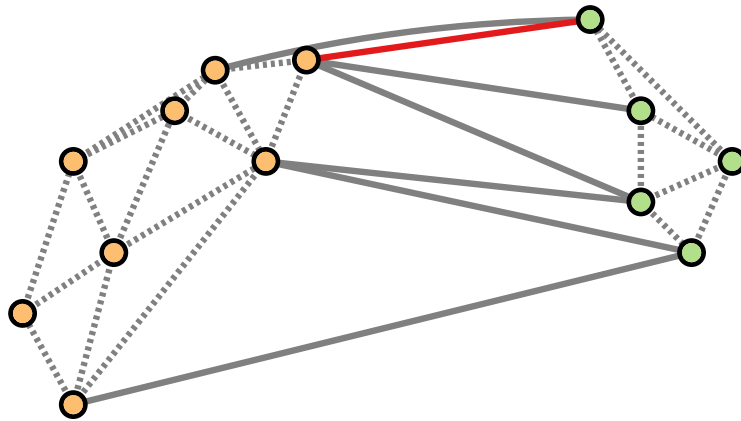
- Delaunay Triangulierung (DT) enthält diese Kante
- DT hat n Kanten, berechenbar in $\mathcal{O}(n \log n)$ Zeit
- Kanten in $\mathcal{G}_{\text{box}}$ bezahlen:

Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



BICHROMATISCHES NÄHESTES PAAR

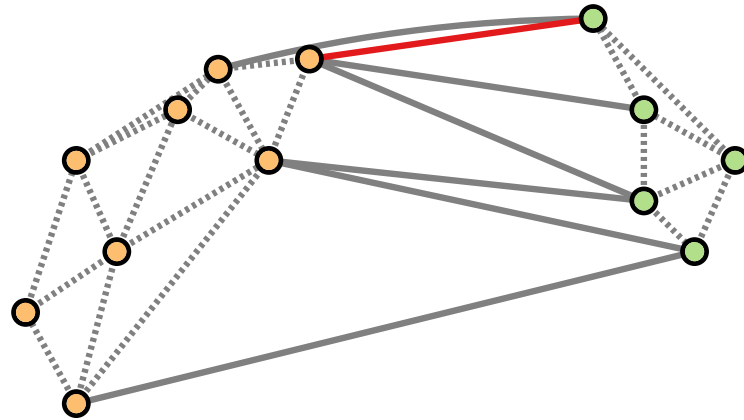
- Delaunay Triangulierung (DT) enthält diese Kante
- DT hat n Kanten, berechenbar in $\mathcal{O}(n \log n)$ Zeit
- Kanten in $\mathcal{G}_{\text{box}}$ bezahlen:
Kante ij bezahlt $c_{ij}(n_i + n_j) \log(n_i + n_j)$ mal.

Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



BICHROMATISCHES NÄHESTES PAAR

- Delaunay Triangulierung (DT) enthält diese Kante
- DT hat n Kanten, berechenbar in $\mathcal{O}(n \log n)$ Zeit
- Kanten in $\mathcal{G}_{\text{box}}$ bezahlen:
Kante ij bezahlt
 $c_{ij}(n_i + n_j) \log(n_i + n_j)$ mal.

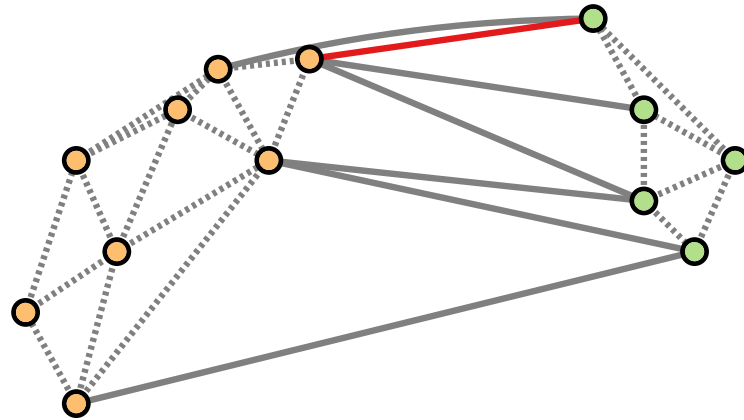
Insgesamt

Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



BICHROMATISCHES NÄHESTES PAAR

- Delaunay Triangulierung (DT) enthält diese Kante
- DT hat n Kanten, berechenbar in $\mathcal{O}(n \log n)$ Zeit
- Kanten in $\mathcal{G}_{\text{box}}$ bezahlen:
Kante ij bezahlt
 $c_{ij}(n_i + n_j) \log(n_i + n_j)$ mal.

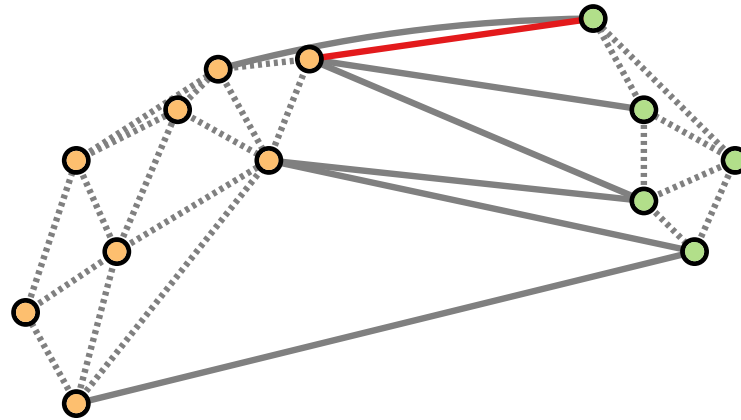
Insgesamt $\mathcal{O}(n \log n)$

Box Graph $\mathcal{G}_{\text{box}}$



Paare von schweren Boxen.

- Das sind alles Kerne.
- Sind sie im **gleichen** Cluster?



BICHROMATISCHES NÄHESTES PAAR

- Delaunay Triangulierung (DT) enthält diese Kante
- DT hat n Kanten, berechenbar in $\mathcal{O}(n \log n)$ Zeit
- Kanten in $\mathcal{G}_{\text{box}}$ bezahlen:
Kante ij bezahlt
 $c_{ij}(n_i + n_j) \log(n_i + n_j)$ mal.

Insgesamt $\mathcal{O}(n \log n)$

weil $\sum_{ij \text{ ist Kante}} n_i \leq 22n_i$.

Ergebnis



Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D	d D	
DBSCAN	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$
HDBSCAN	$\mathcal{O}(n \log n)$ expected	$\times$	

Ergebnis



Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D	dD	
DBSCAN	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$
HDBSCAN	$\mathcal{O}(n \log n)$ expected	$\times$	



1. Erstelle $\mathcal{G}_{\text{box}}$
2. Finde Kerne
3. Verbinde Cluster
- (4. Weise Randpunkte zu.)

Ergebnis



Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D	dD	
DBSCAN	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$
HDBSCAN	$\mathcal{O}(n \log n)$ expected	$\times$	



1. Erstelle $\mathcal{G}_{\text{box}}$

2. Finde Kerne

3. Verbinde Cluster

(4. Weise Randpunkte zu.)

BICHROMATISCHES NÄHESTES PAAR statt
Delaunay Triangulierung.

Ergebnis



Annahme: ε frei, k feste Konstante, Euklidische Abstände

	2D	dD	
DBSCAN	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \gamma})$	$\gamma > 0$
HDBSCAN	$\mathcal{O}(n \log n)$ expected	$\times$	

1. Erstelle $\mathcal{G}_{\text{box}}$

2. Finde Kerne

3. Verbinde Cluster

(4. Weise Randpunkte zu.)

BICHROMATISCHES NÄHESTES PAAR statt
Delaunay Triangulierung.
(Agarwal, Edelsbrunner, Schwarzkopf, 1991)