Computational Geometry
Grafisk Databehandling

1. *Convex hull* – konveks innhyling

2. Nærmeste par av punkter

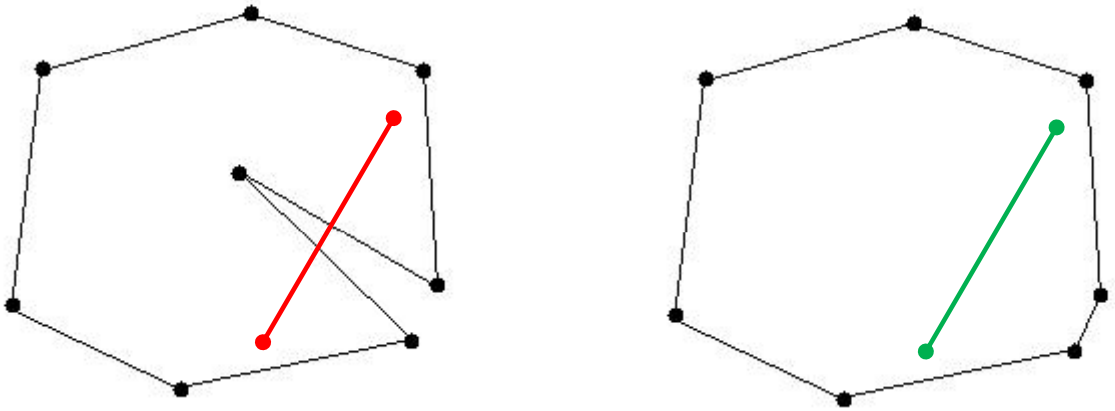
Convex hull
Konveks innhylling

Konveks innhylling

La P være en mengde punkter i et k-dimensjonalt rom, $P \in \mathbf{R}^k$.

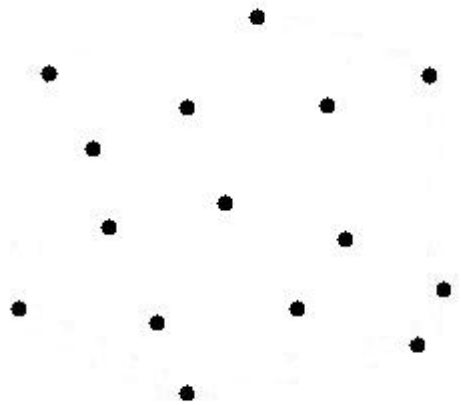
(Vi skal for enkelthets skyld bare se på $k = 2$.)

Definisjon En mengde $Q \in \mathbf{R}^k$ er *konveks* dersom for alle punkter $q_1, q_2 \in Q$ linjesegmentet $\overline{q_1 q_2}$ ligger i Q.

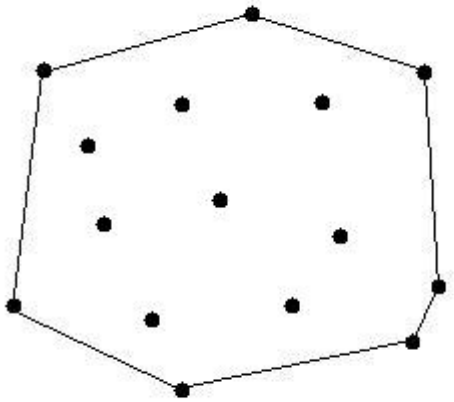


Definisjon Den *konvekse innhyllingen* til en mengde punkter $P \in \mathbf{R}^k$ er den minste konvekse mengde Q som inneholder punktene i P.

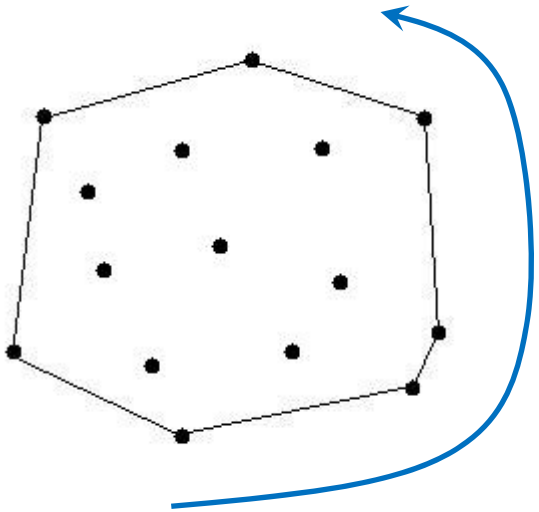
Konveks innhylling



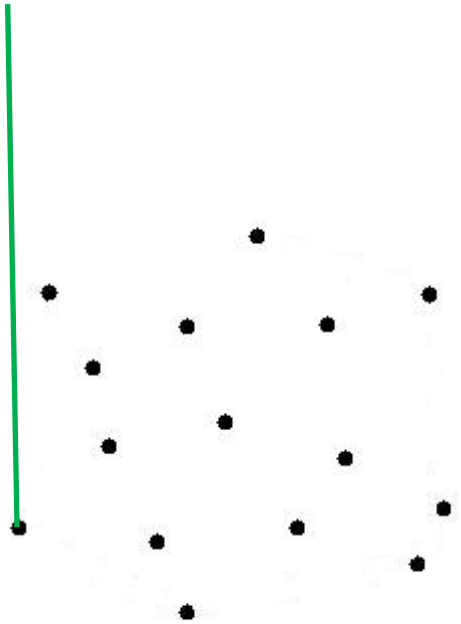
Konveks innhylling



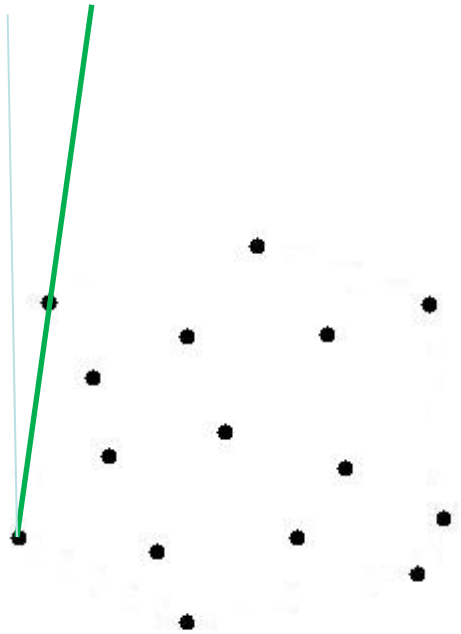
Konveks innhylling



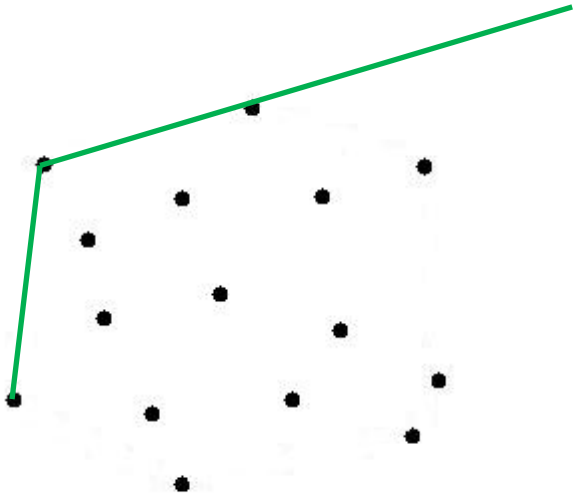
Konveks innhylling



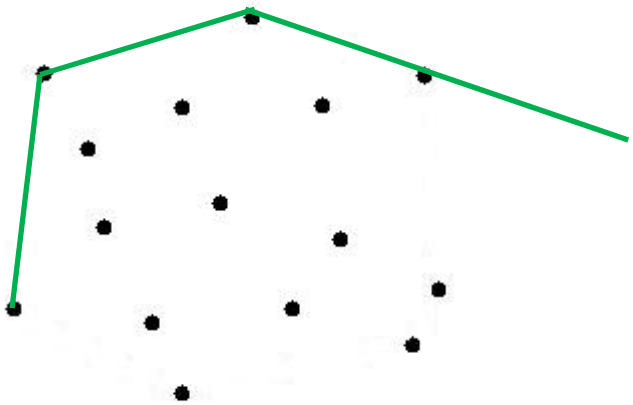
Konveks innhylling



Konveks innhylling

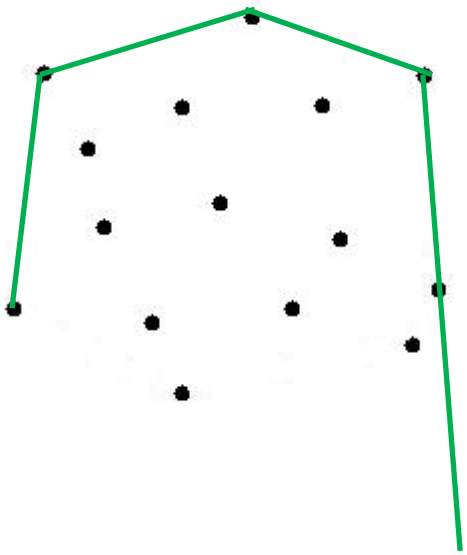


Konveks innhylling



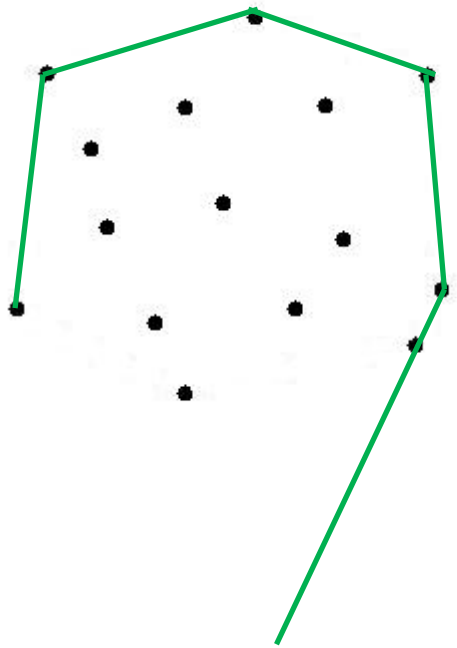
O(*hn*)

Konveks innhylling



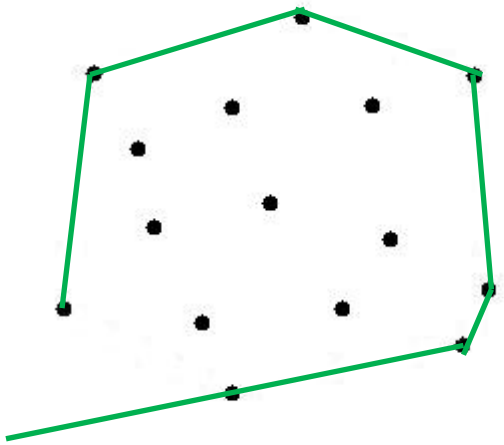
O(*hn*)

Konveks innhylling



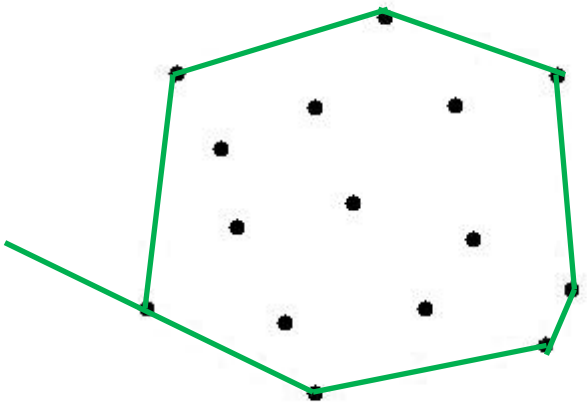
O(*hn*)

Konveks innhylling



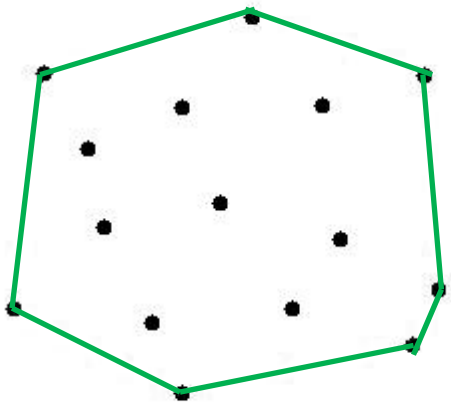
O(*hn*)

Konveks innhylling



O(*hn*)

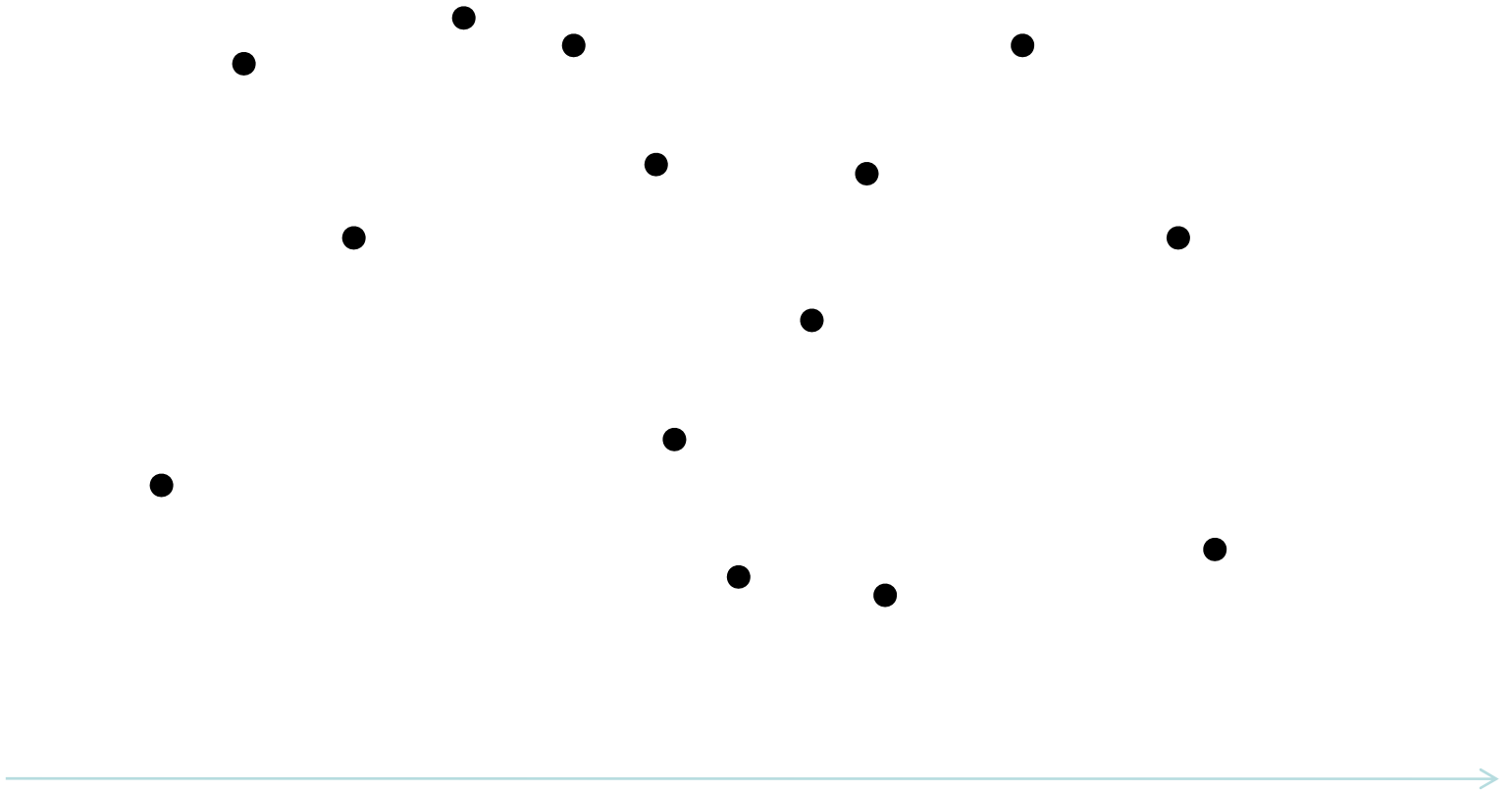
Konveks innhylling



Jarvis' match

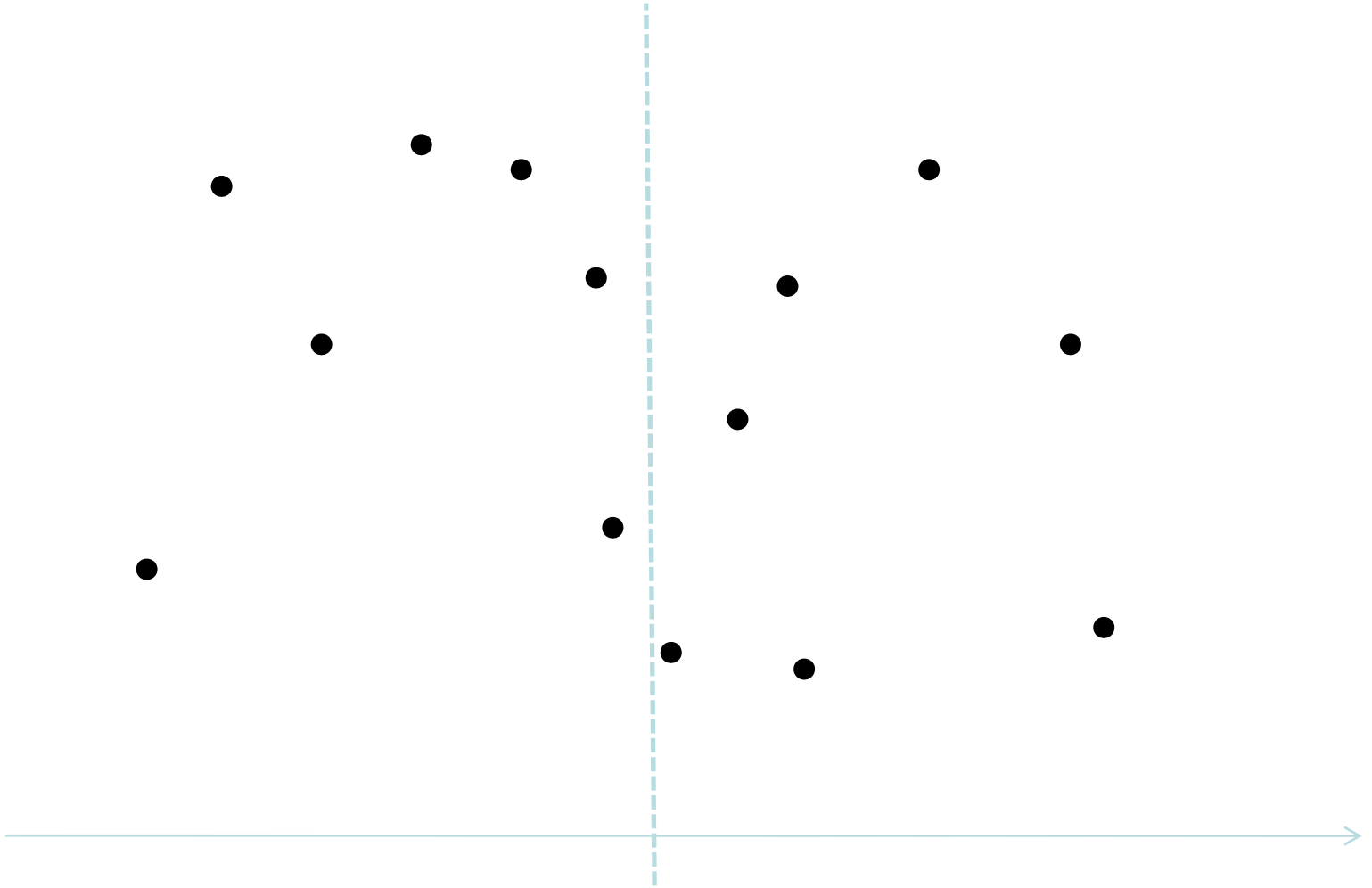
O(*hn*)

Konveks innhylling



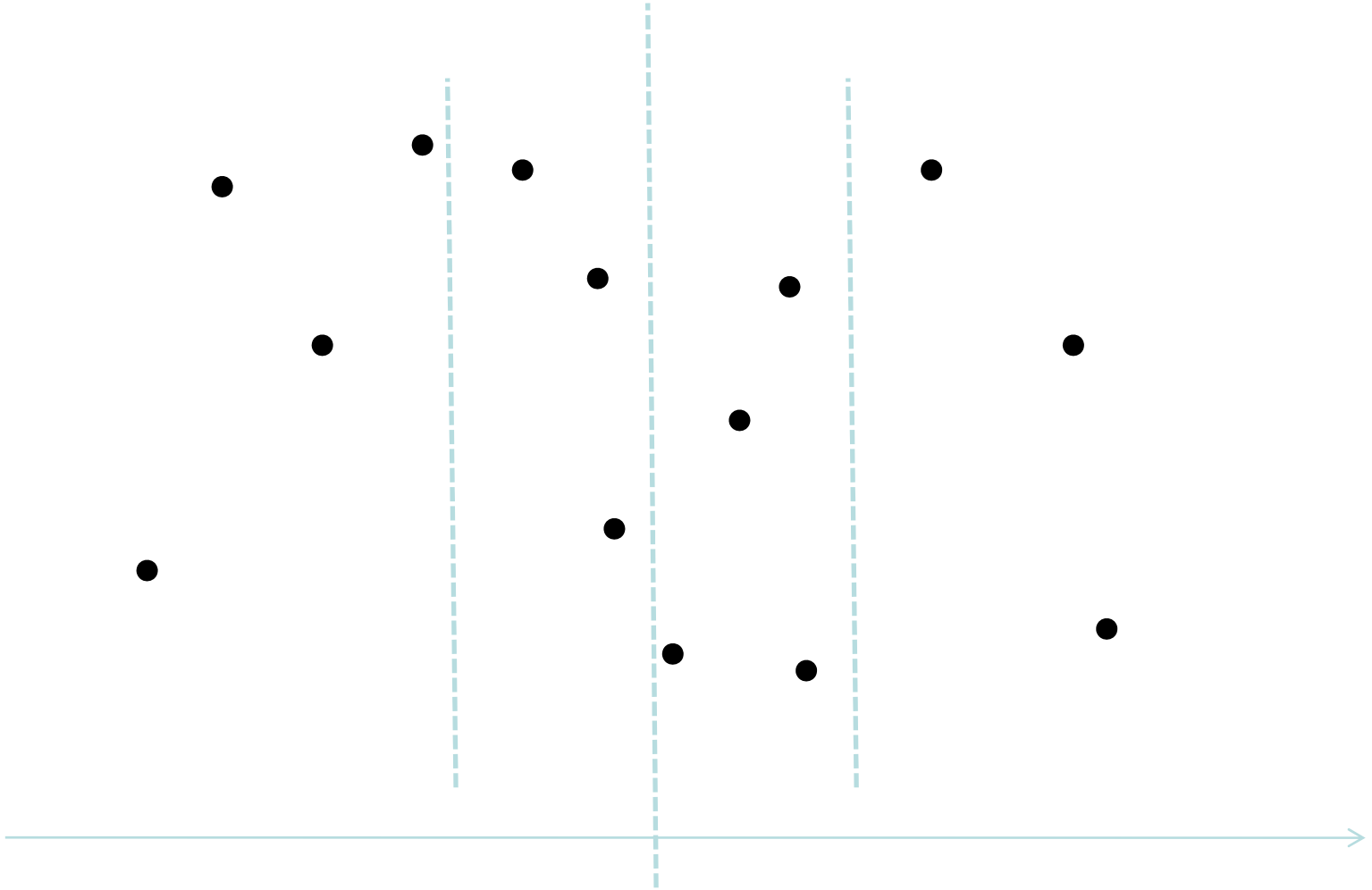
Deler i to ved *x*-median, helt til vi får 1, 2 eller 3 punkter i mengden.

Konveks innhylling



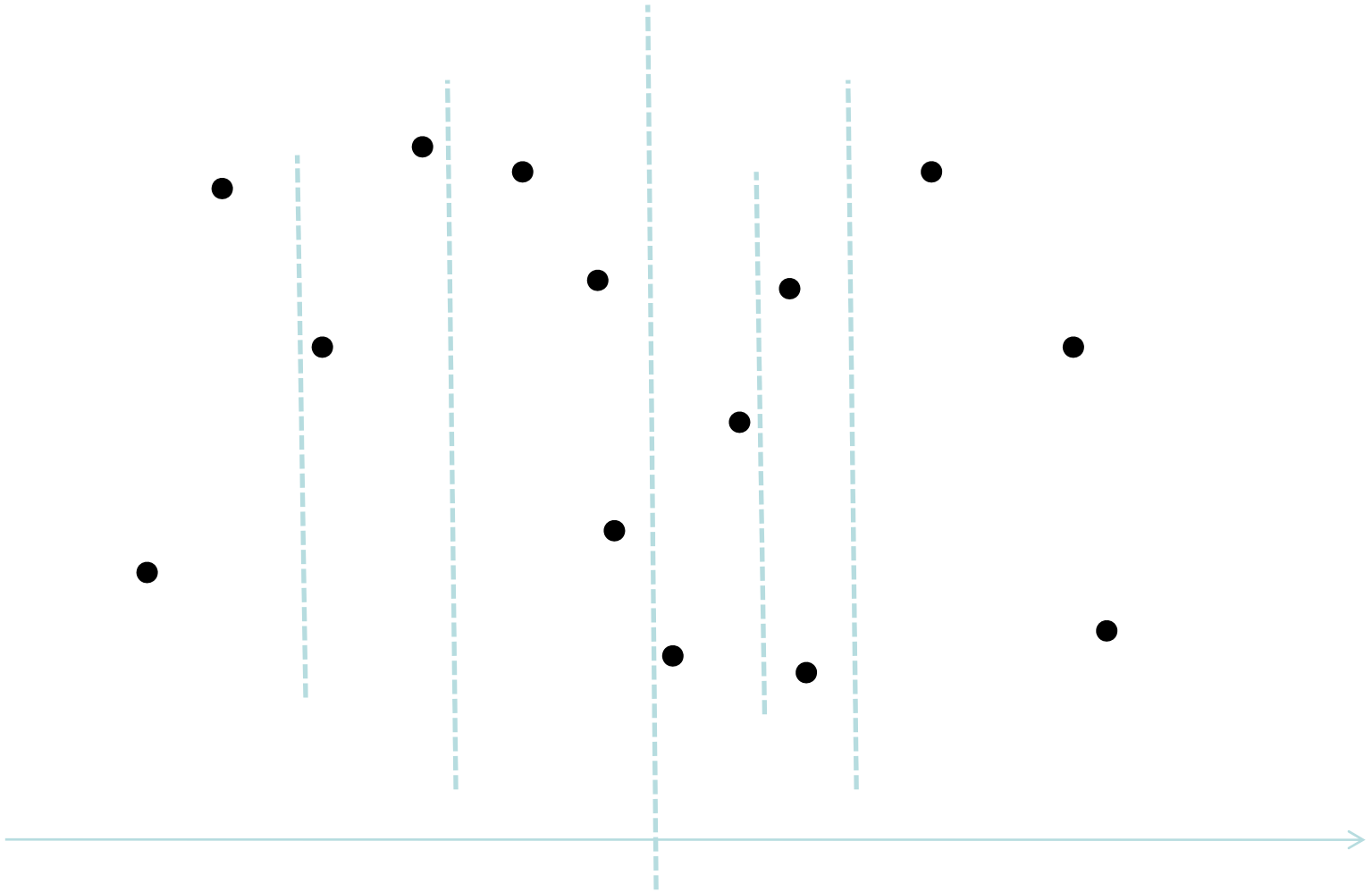
Deler i to ved *x*-median, helt til vi får 1, 2 eller 3 punkter i mengden.

Konveks innhylling



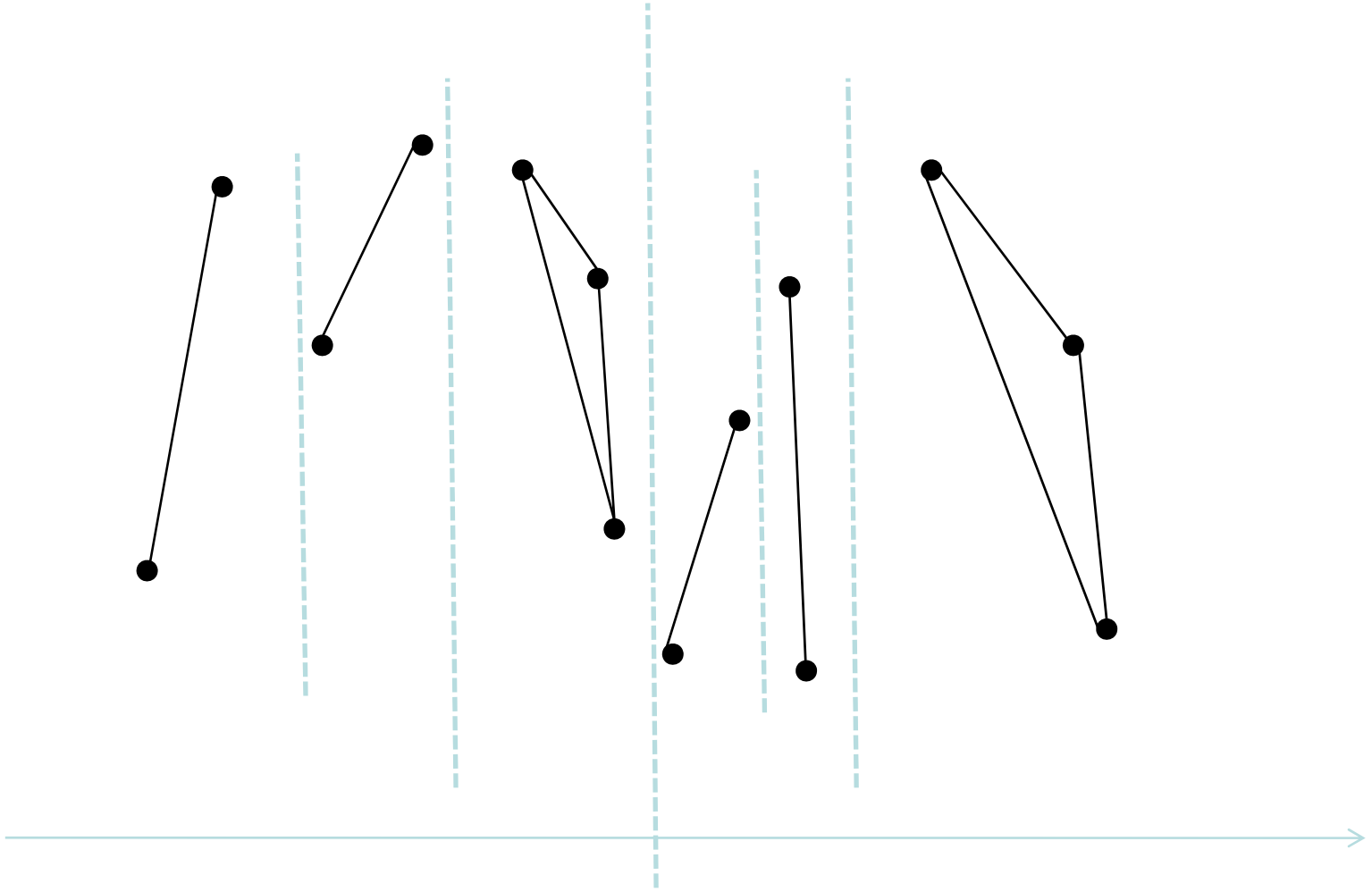
Deler i to ved *x*-median, helt til vi får 1, 2 eller 3 punkter i mengden.

Konveks innhylling



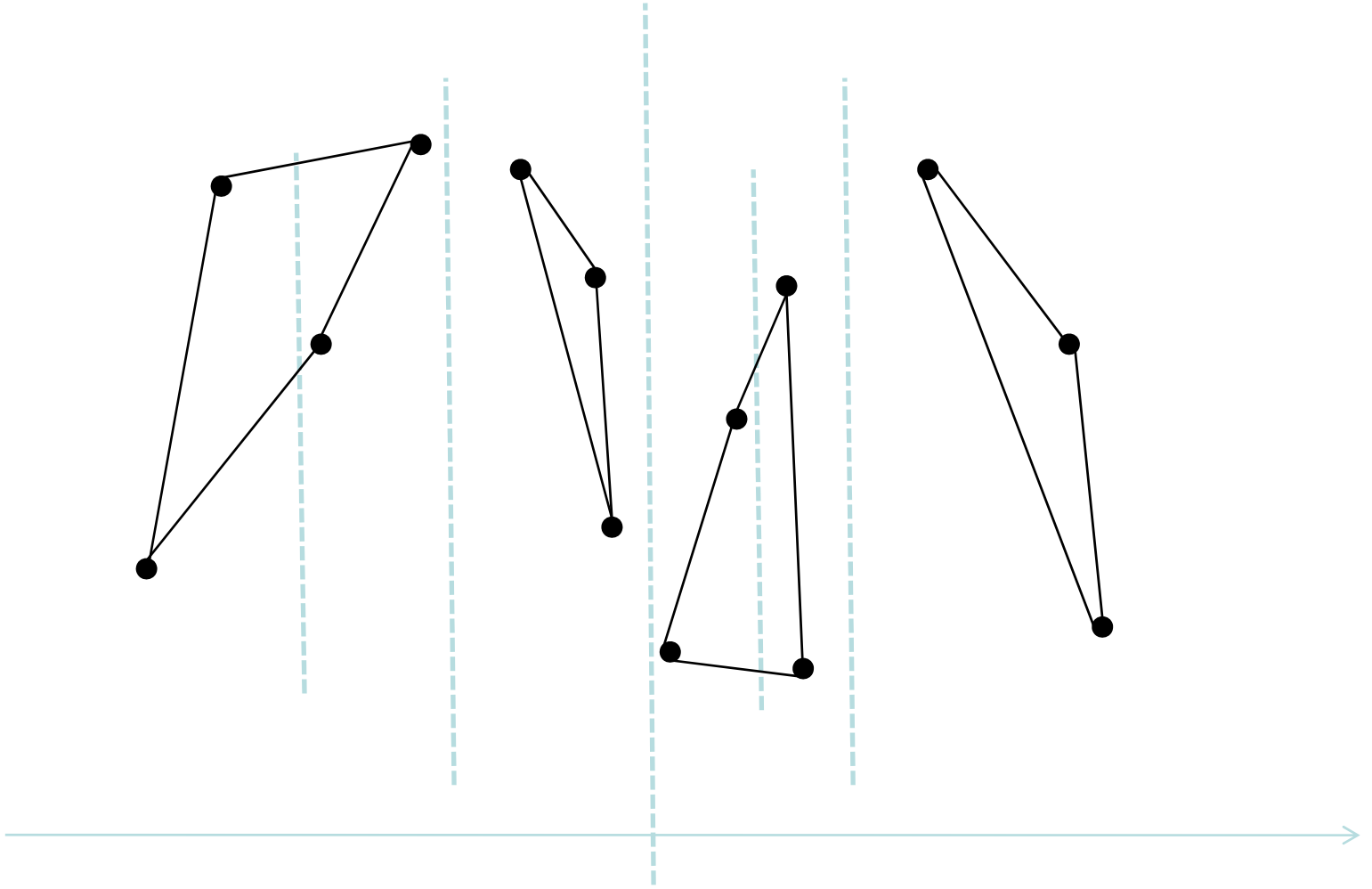
Deler i to ved *x*-median, helt til vi får 1, 2 eller 3 punkter i mengden.

Konveks innhylling



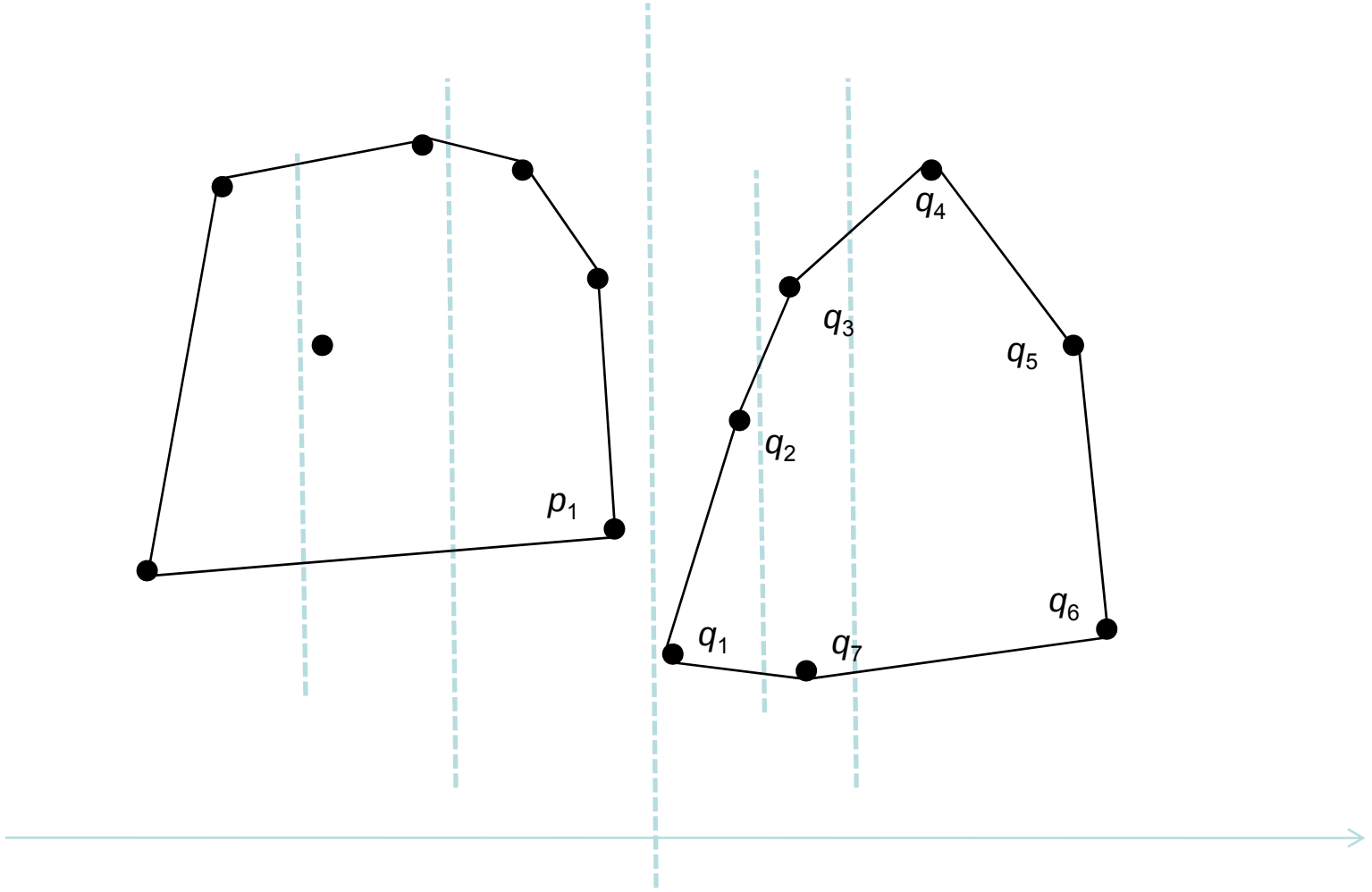
Finner enkle (1, 2 eller 3 punkter) innhyllinger

Konveks innhylling

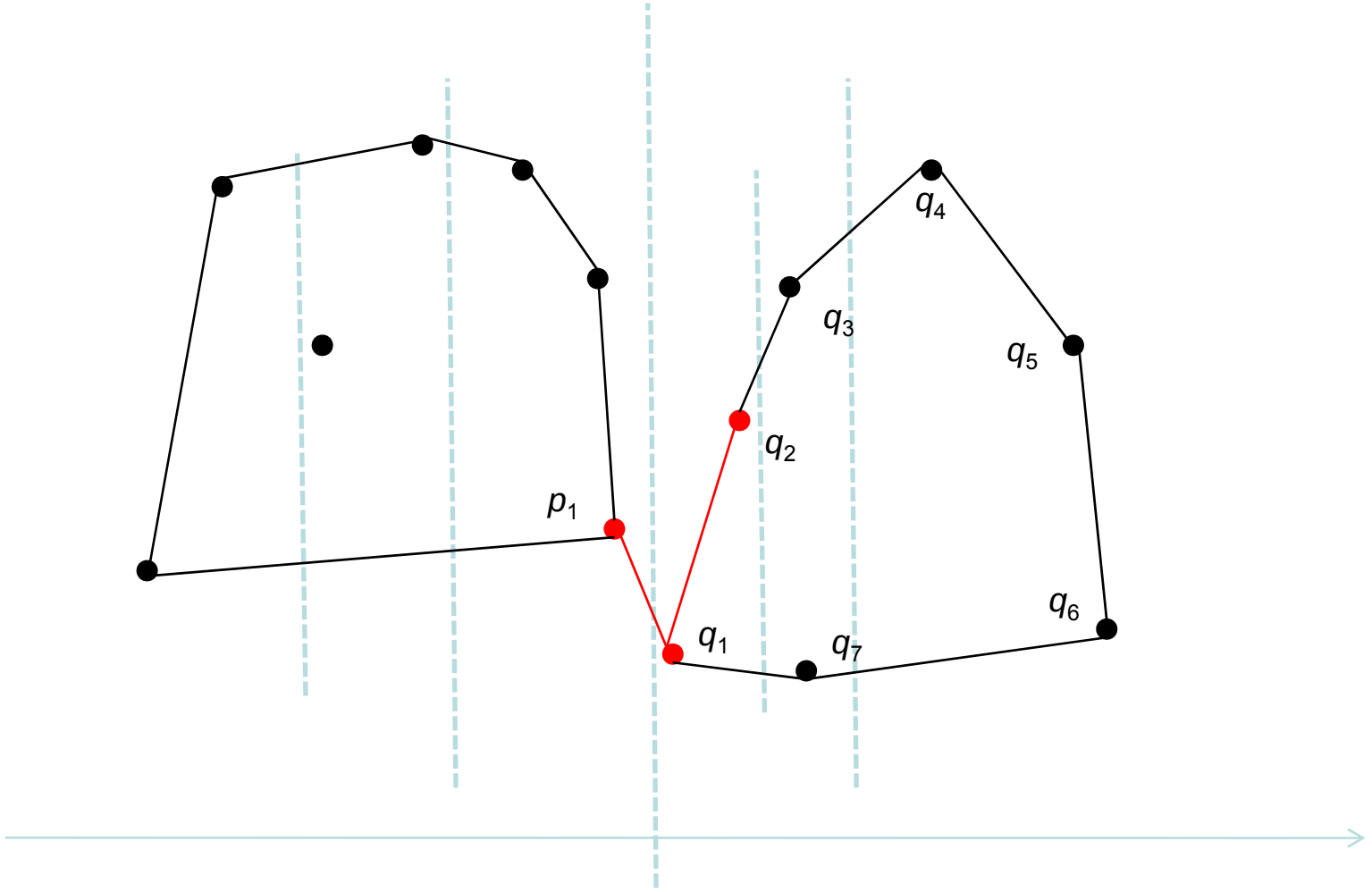


Kobler sammen to og to innhyllinger.

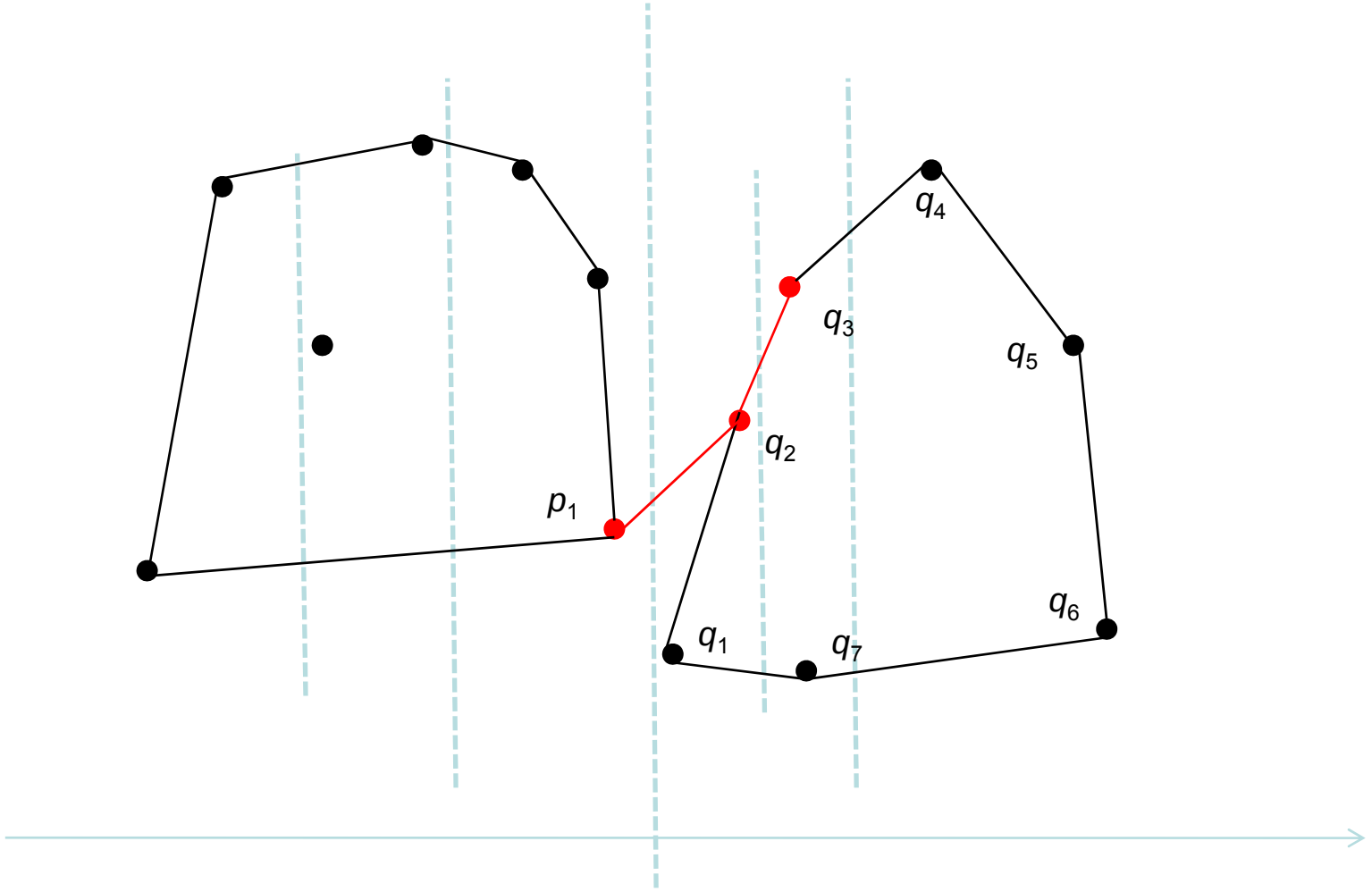
Konveks innhylling



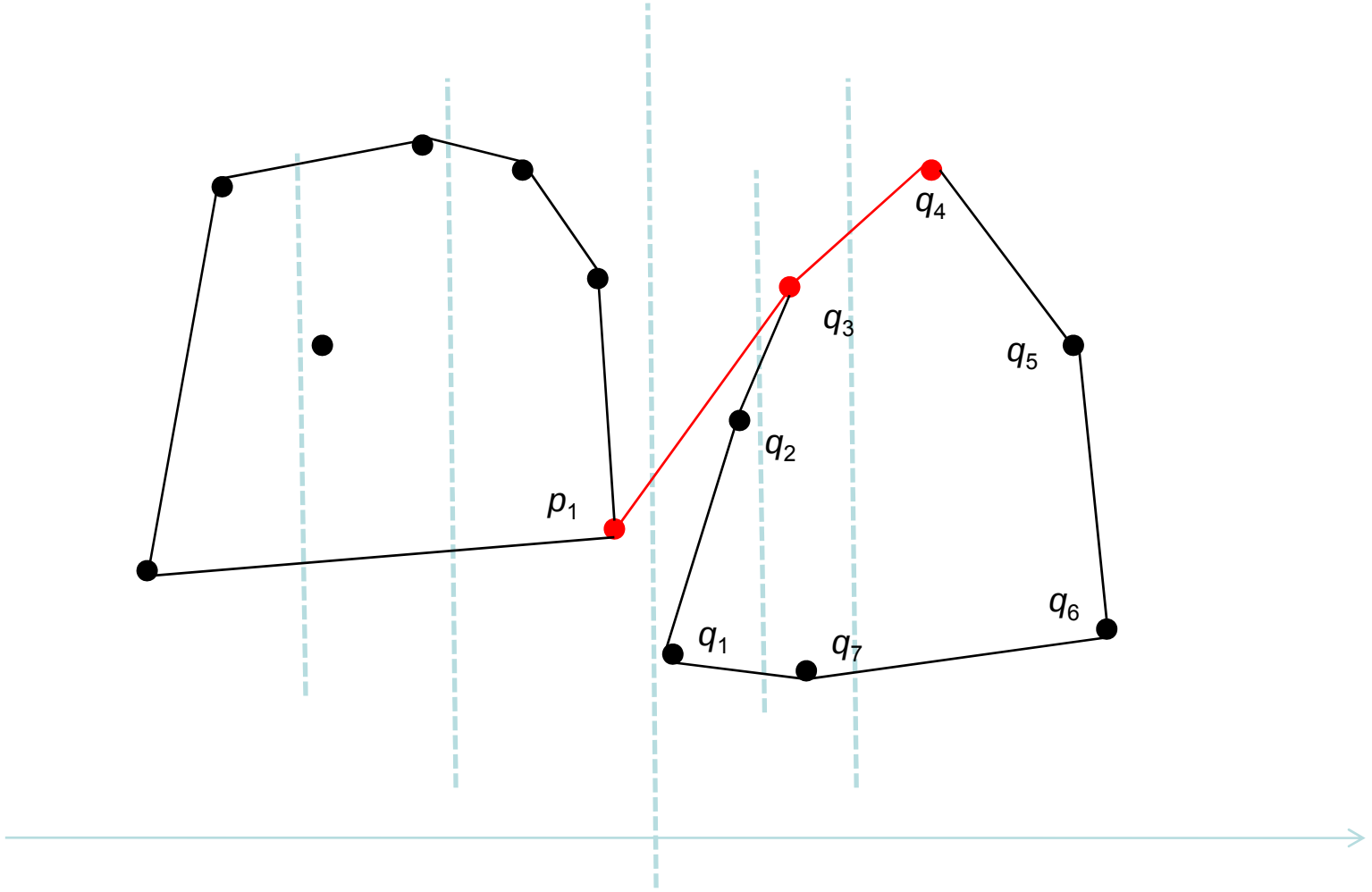
Konveks innhylling



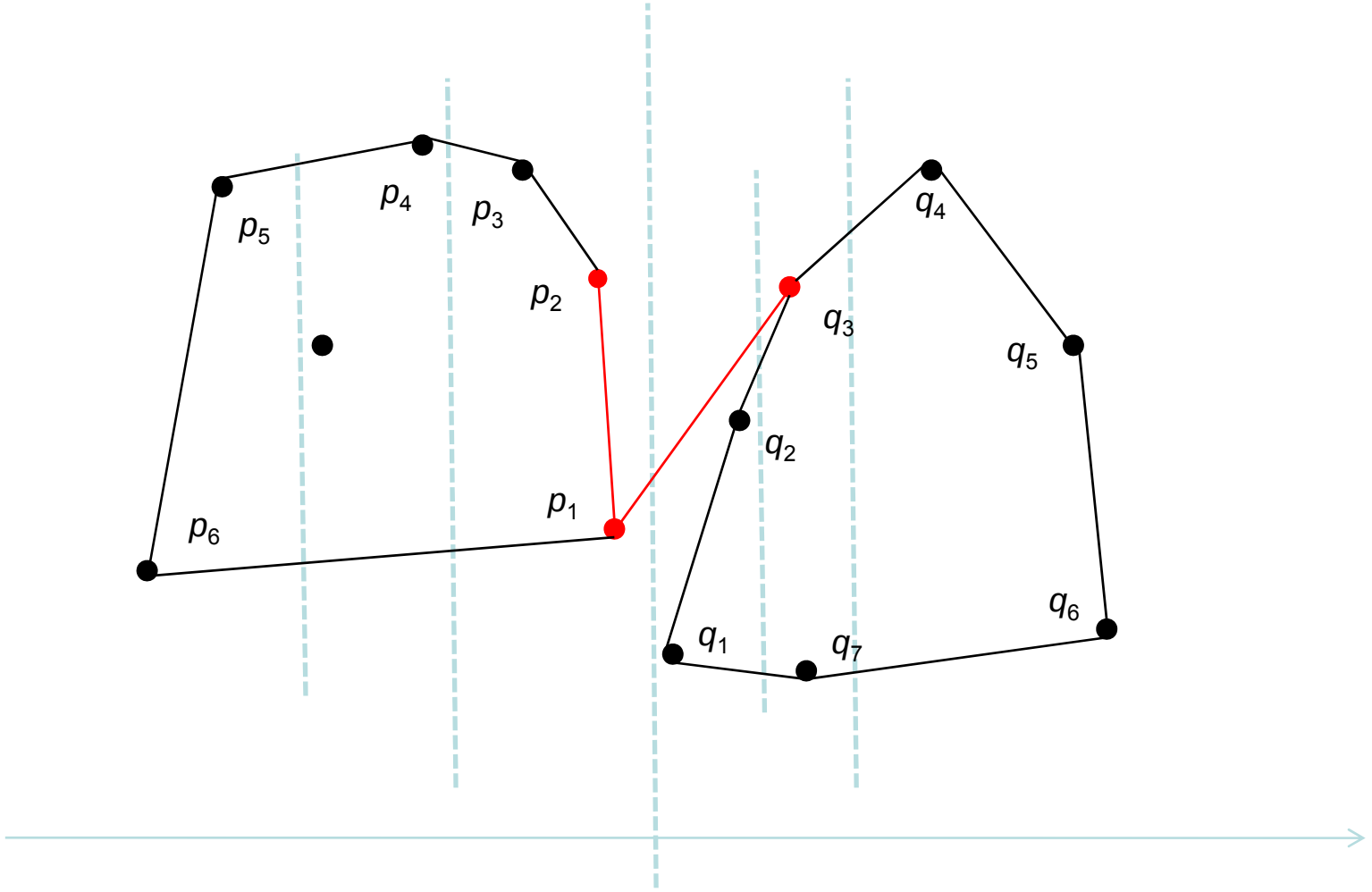
Konveks innhylling



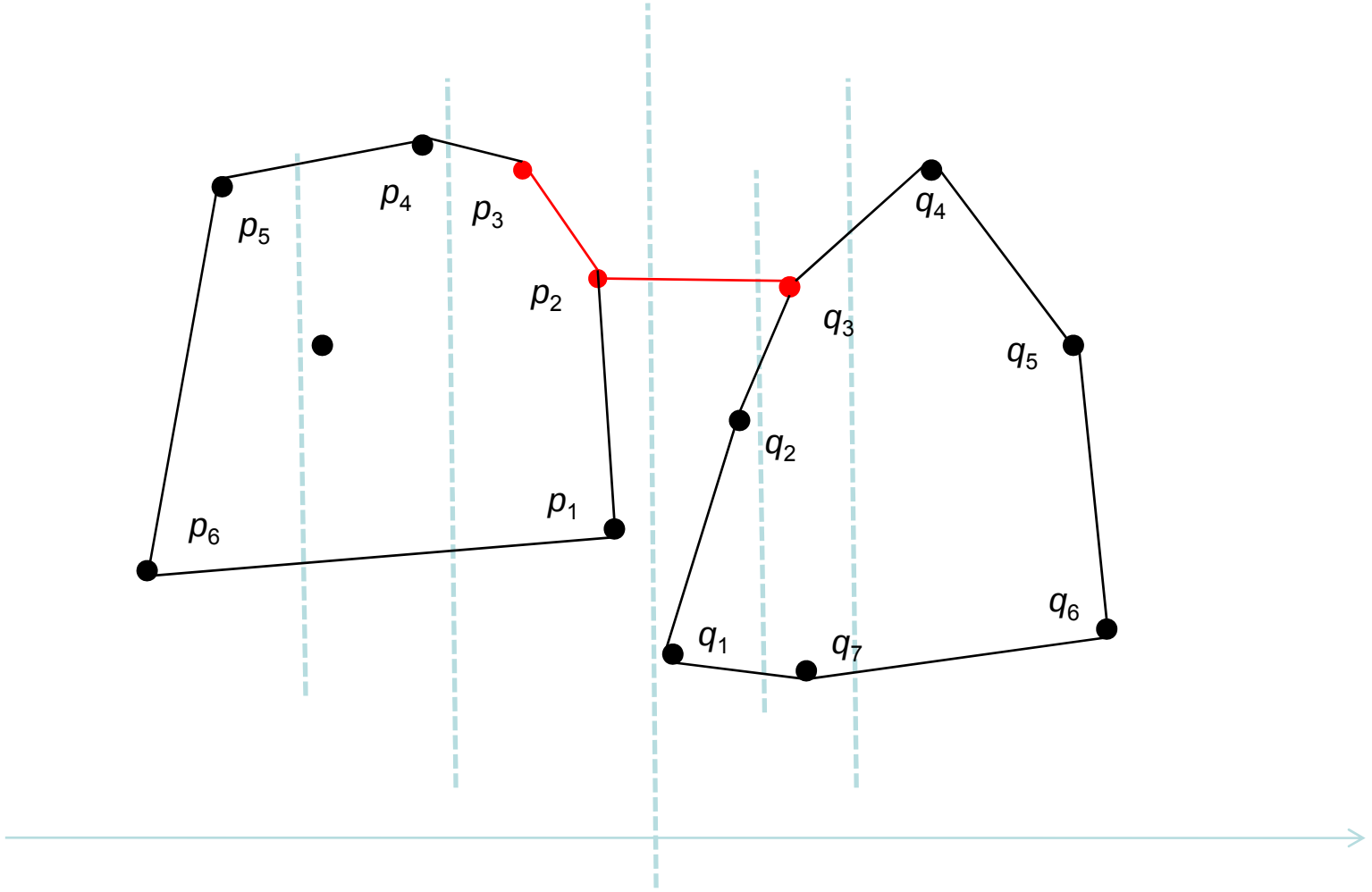
Konveks innhylling



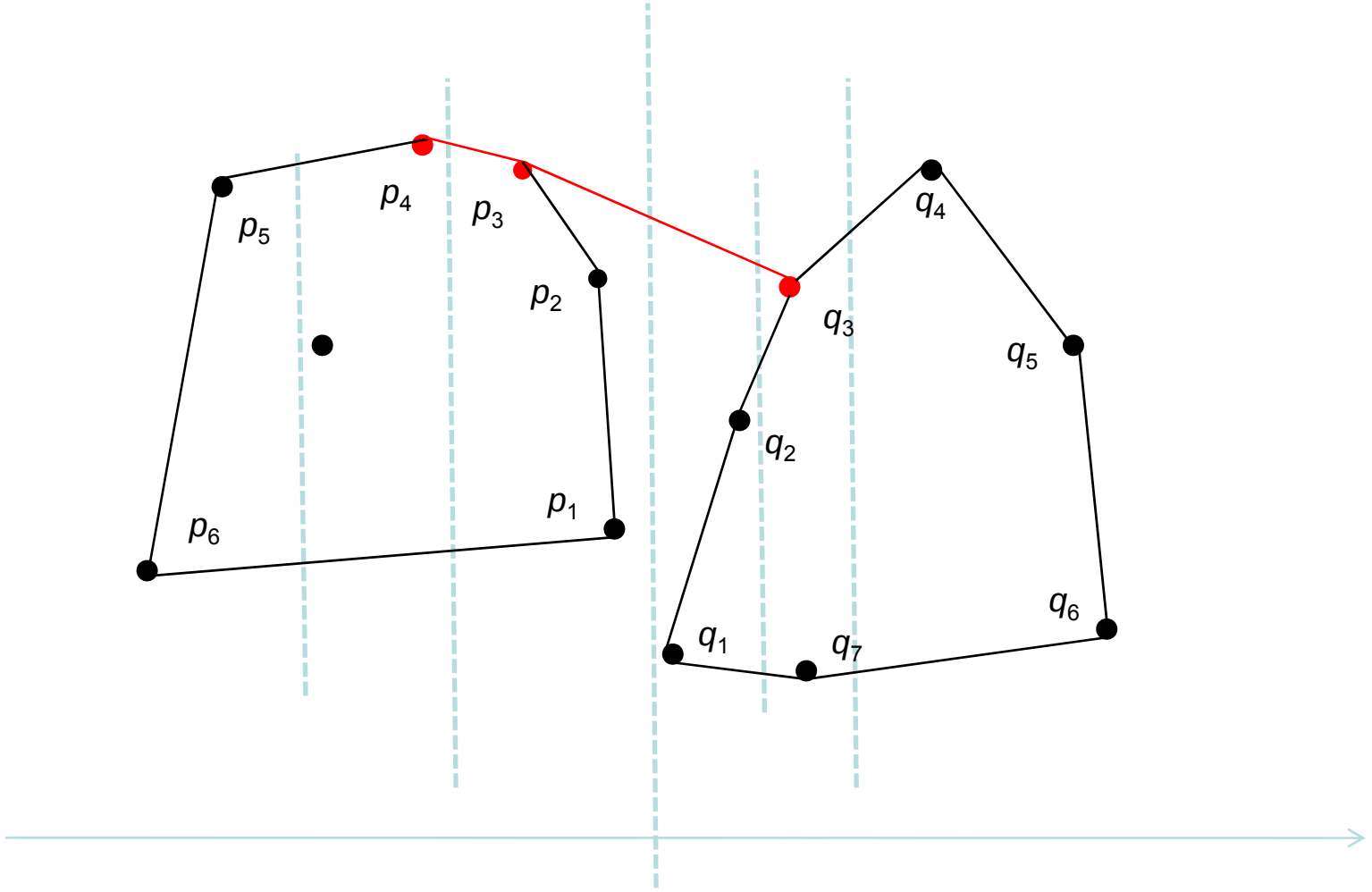
Konveks innhylling



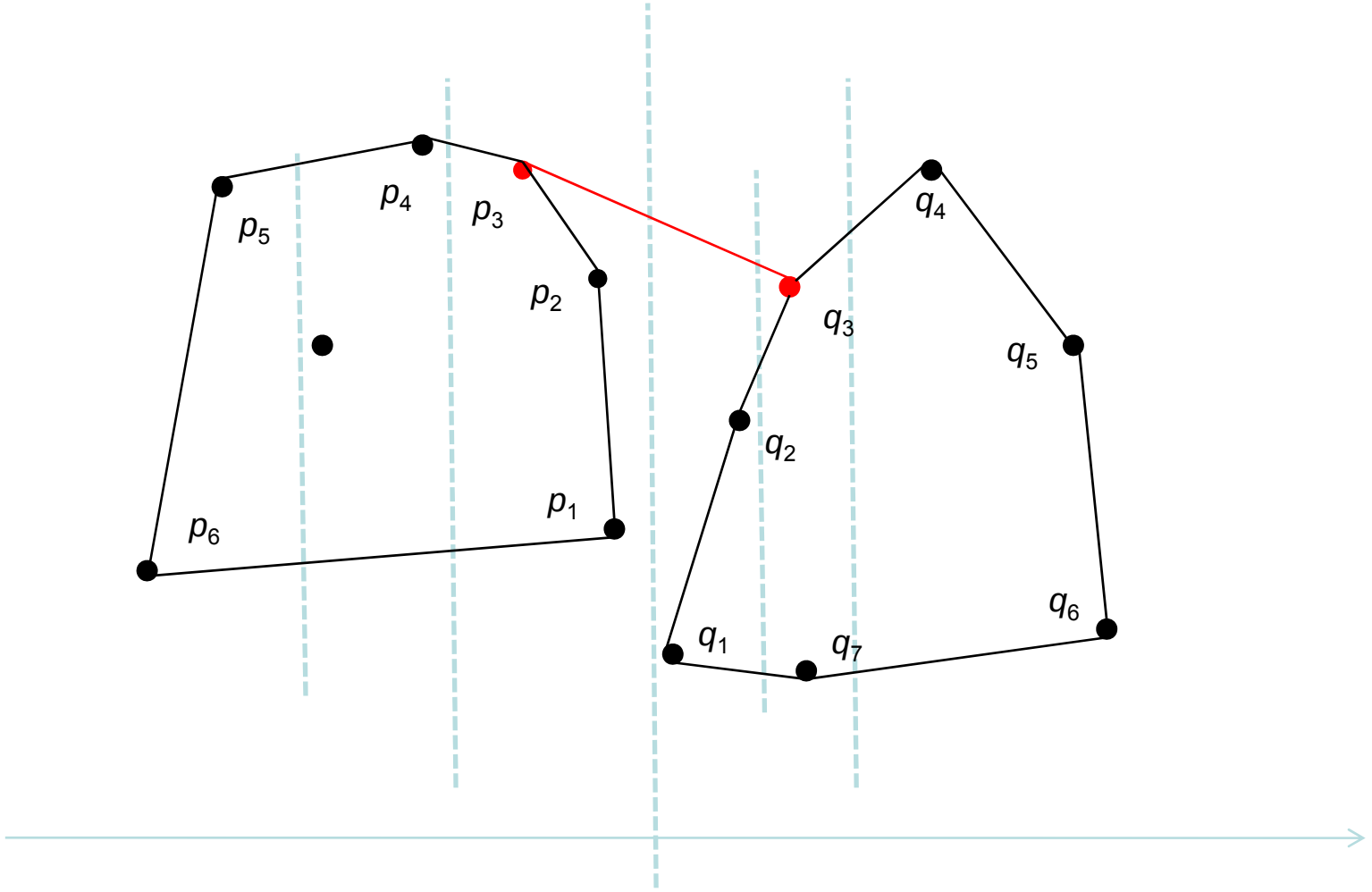
Konveks innhylling



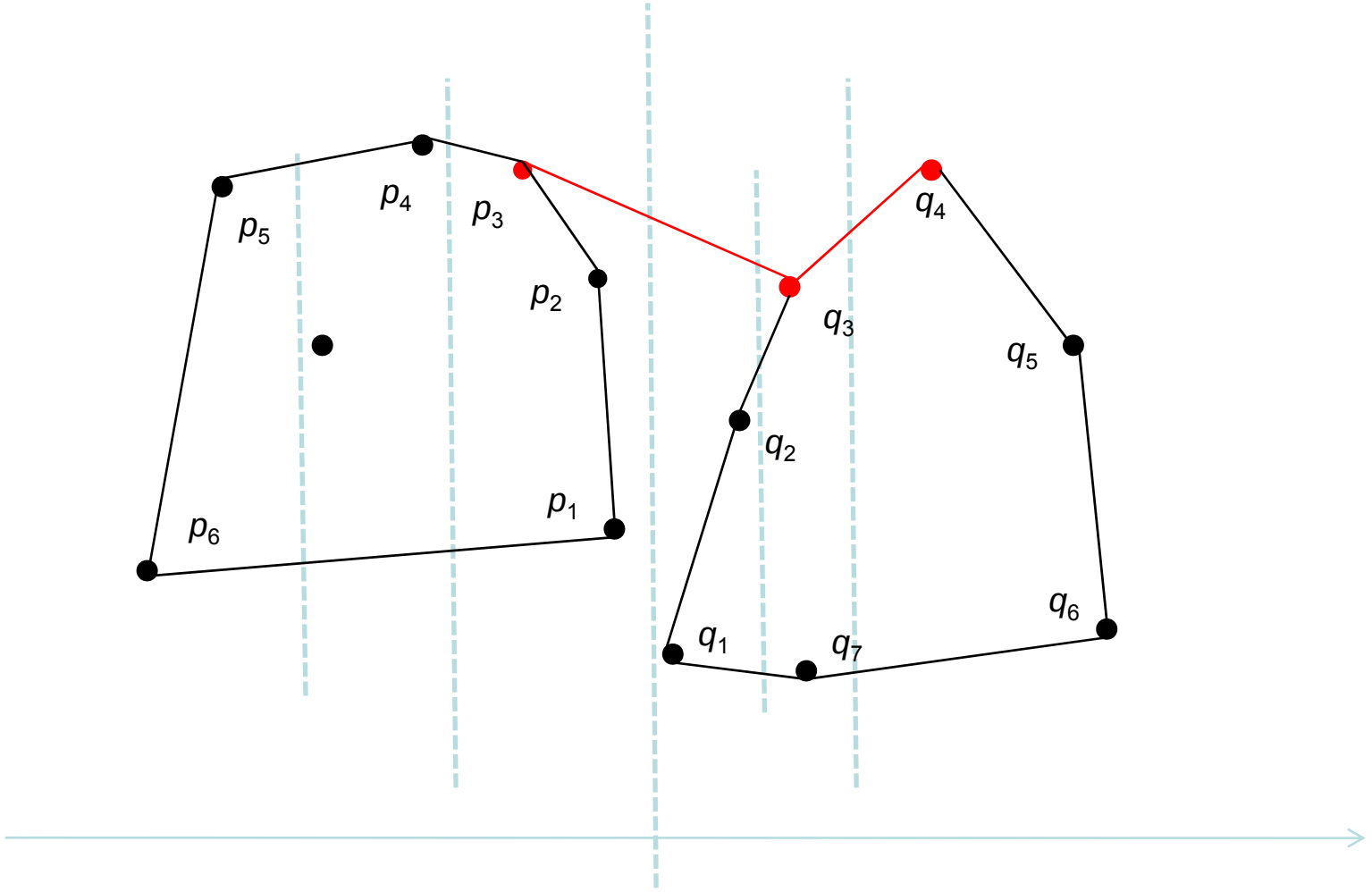
Konveks innhylling



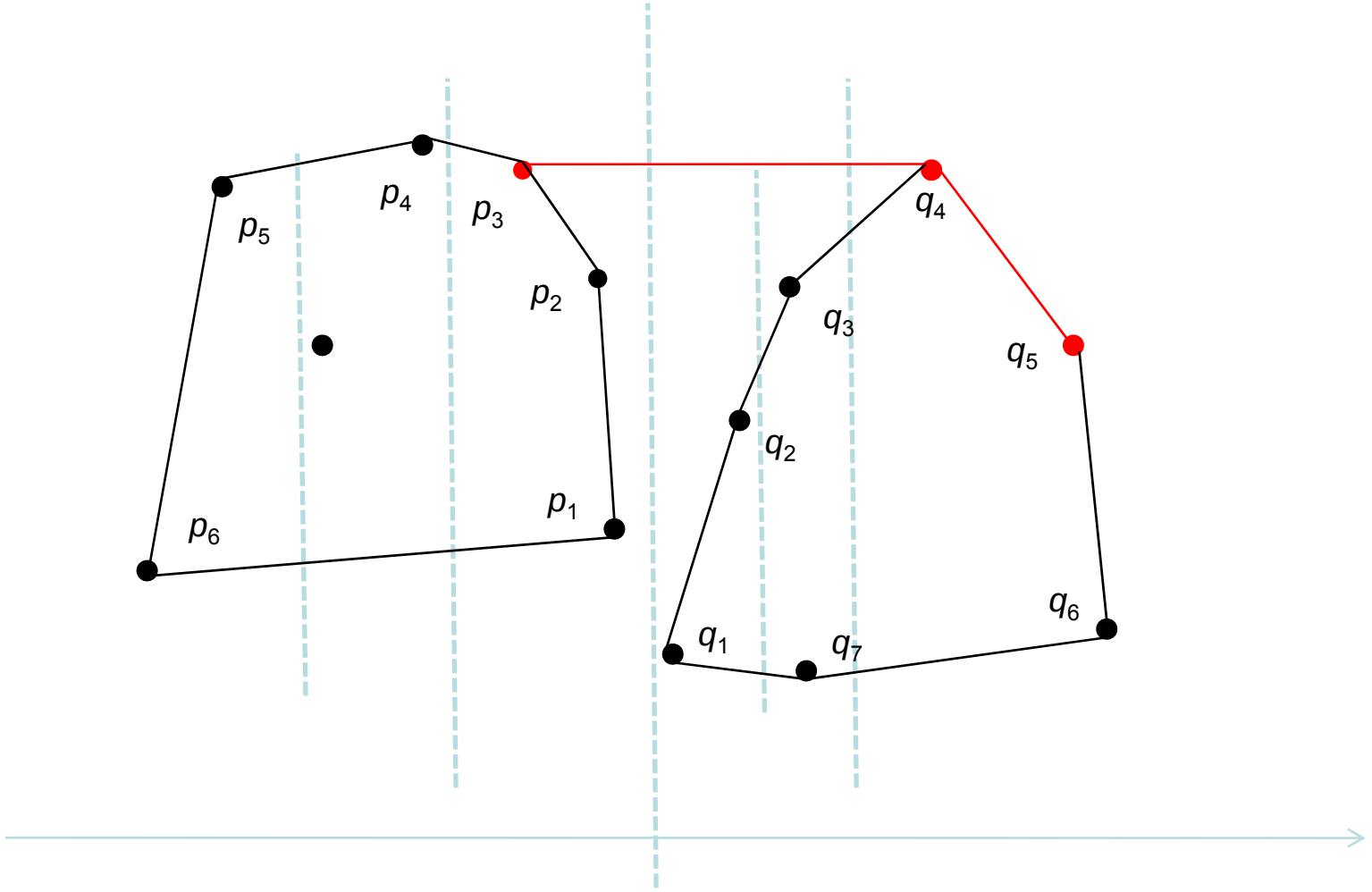
Konveks innhylling



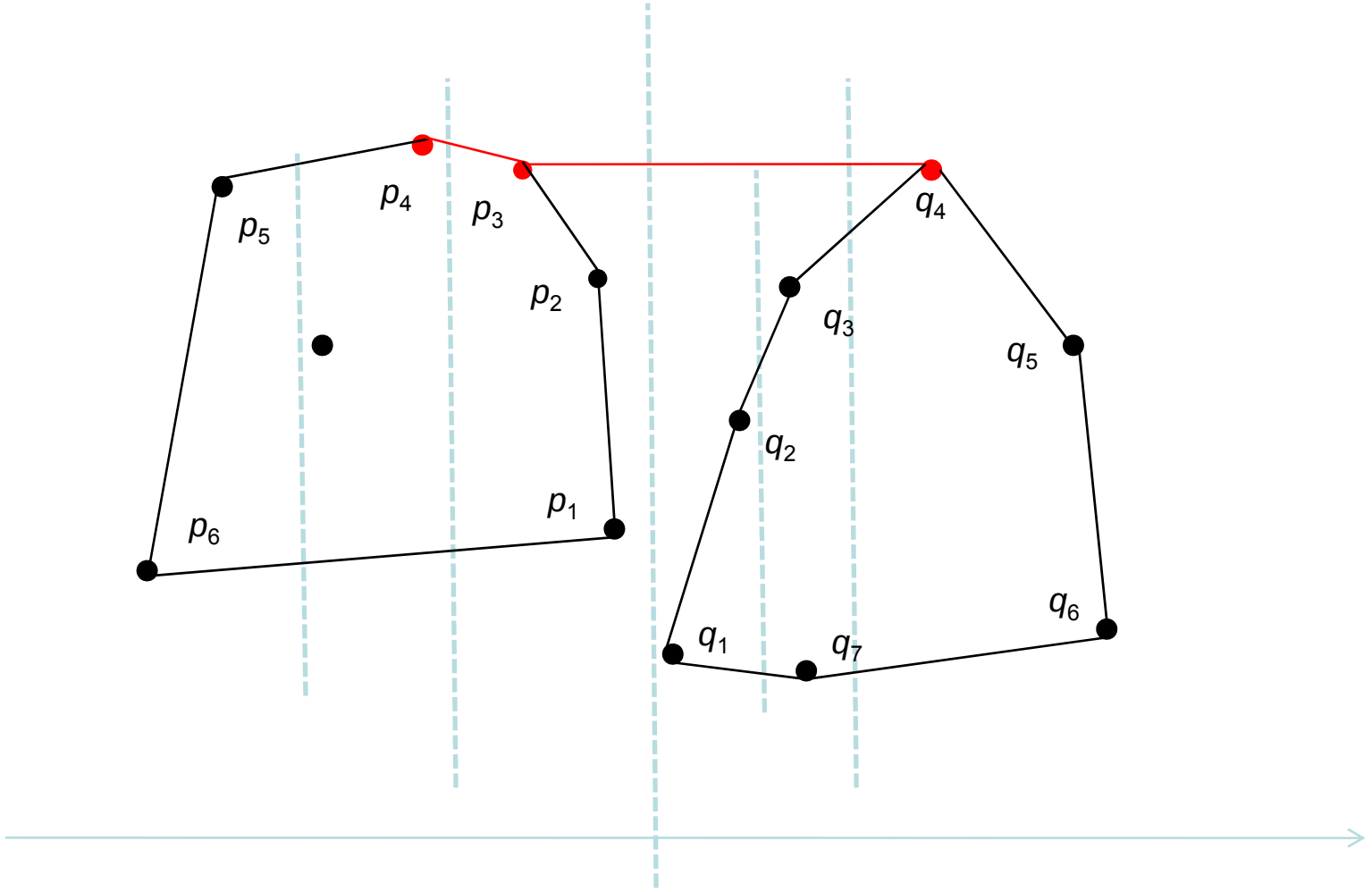
Konveks innhylling



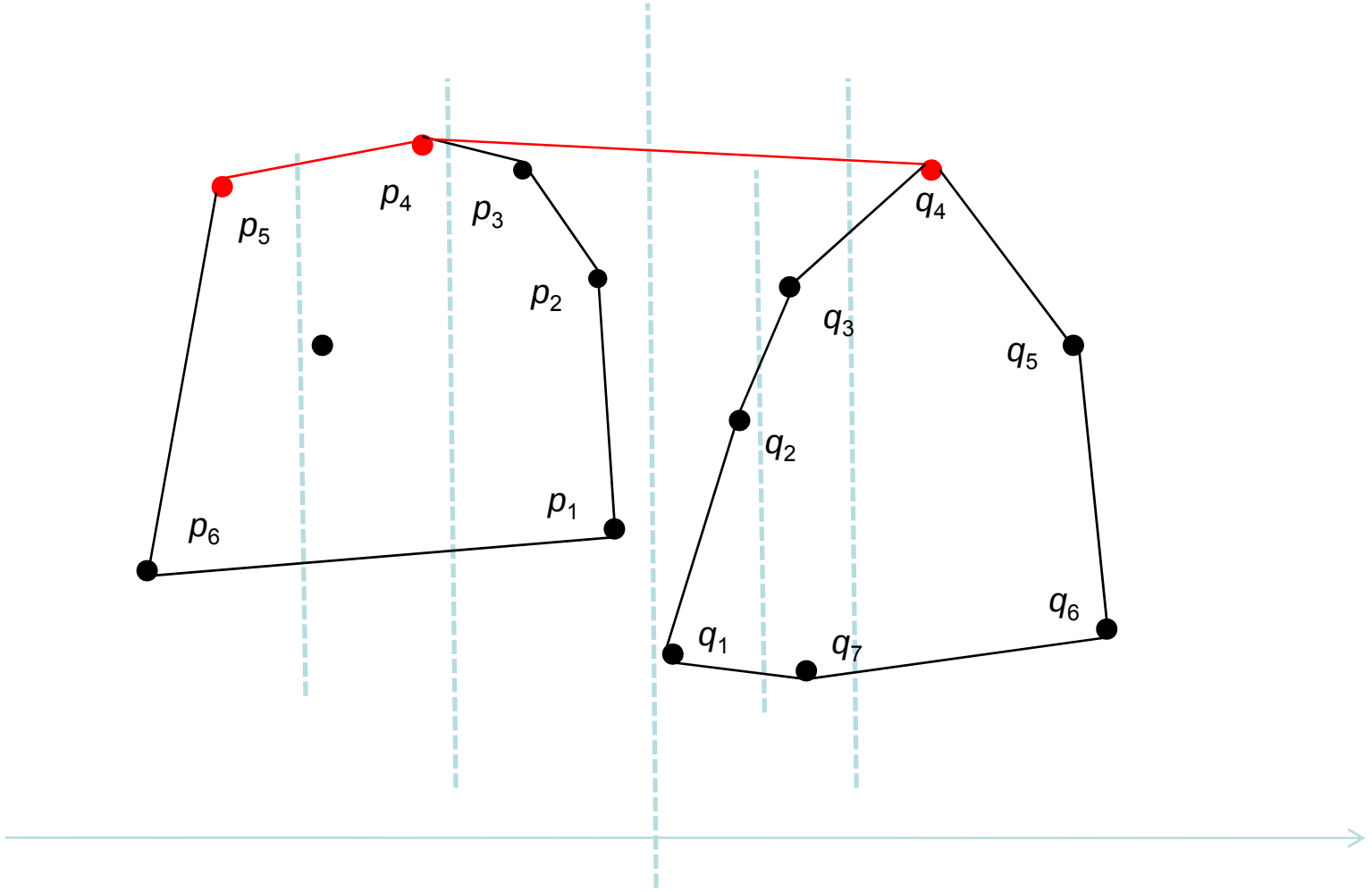
Konveks innhylling



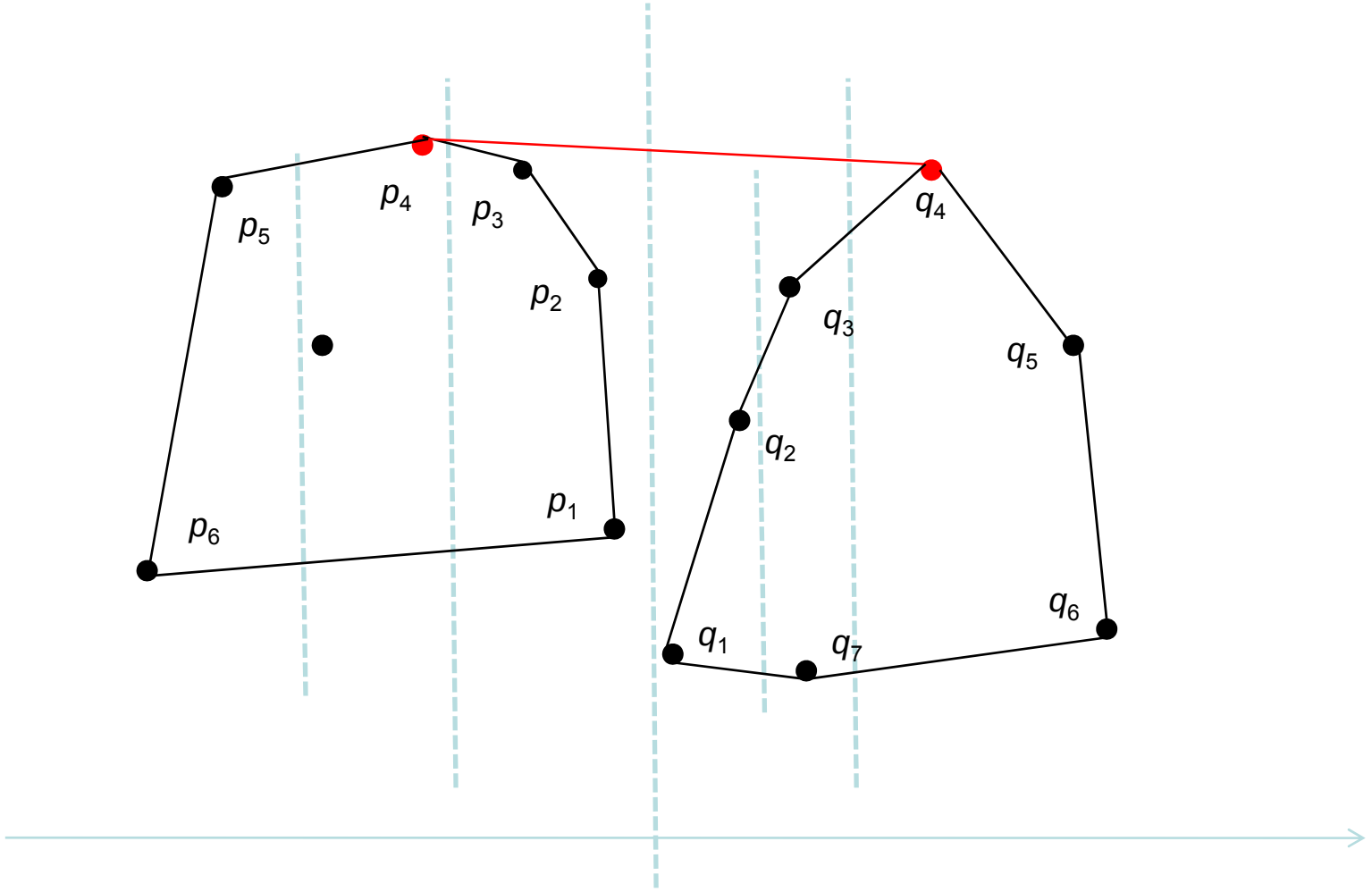
Konveks innhylling



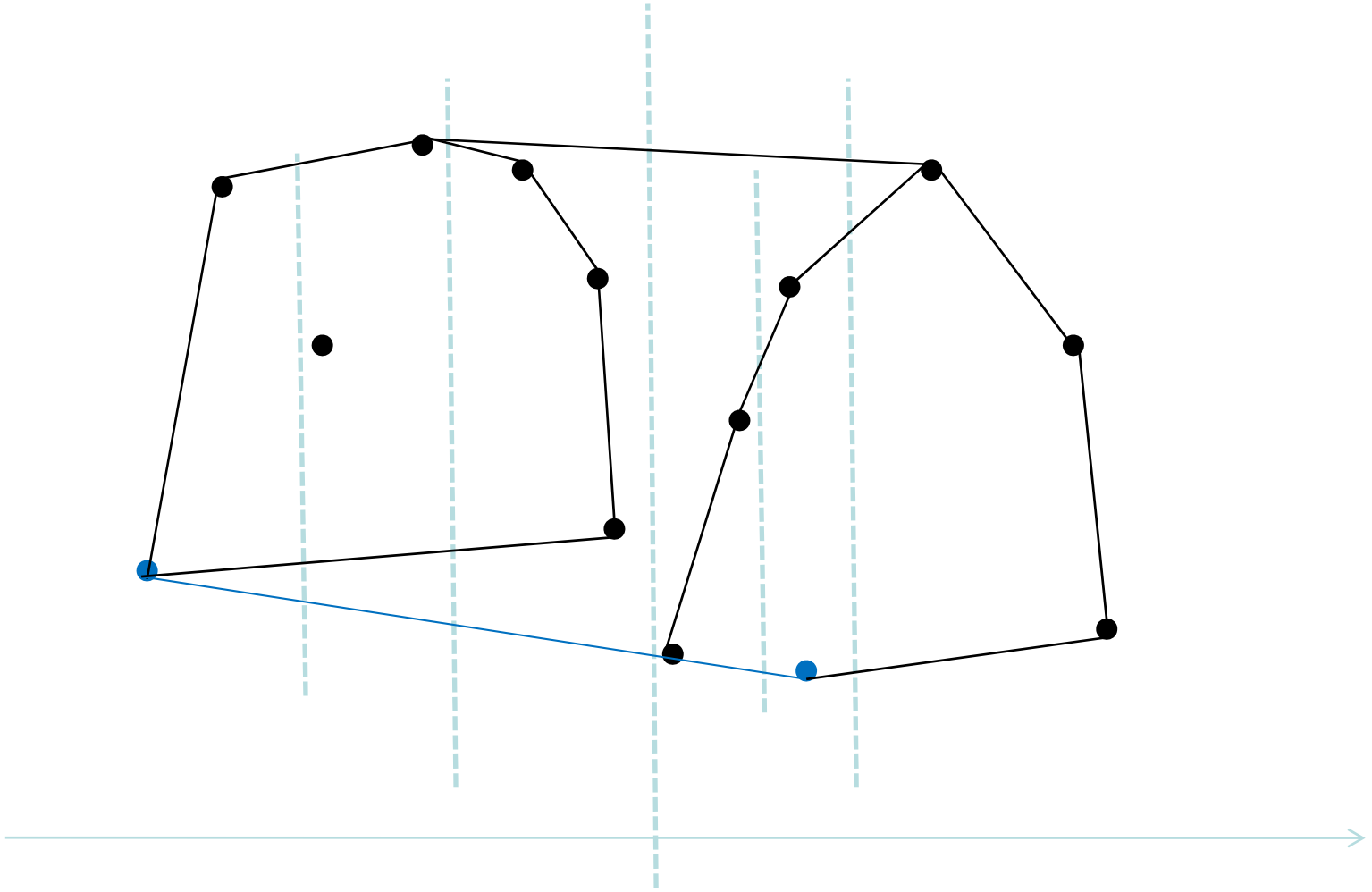
Konveks innhylling



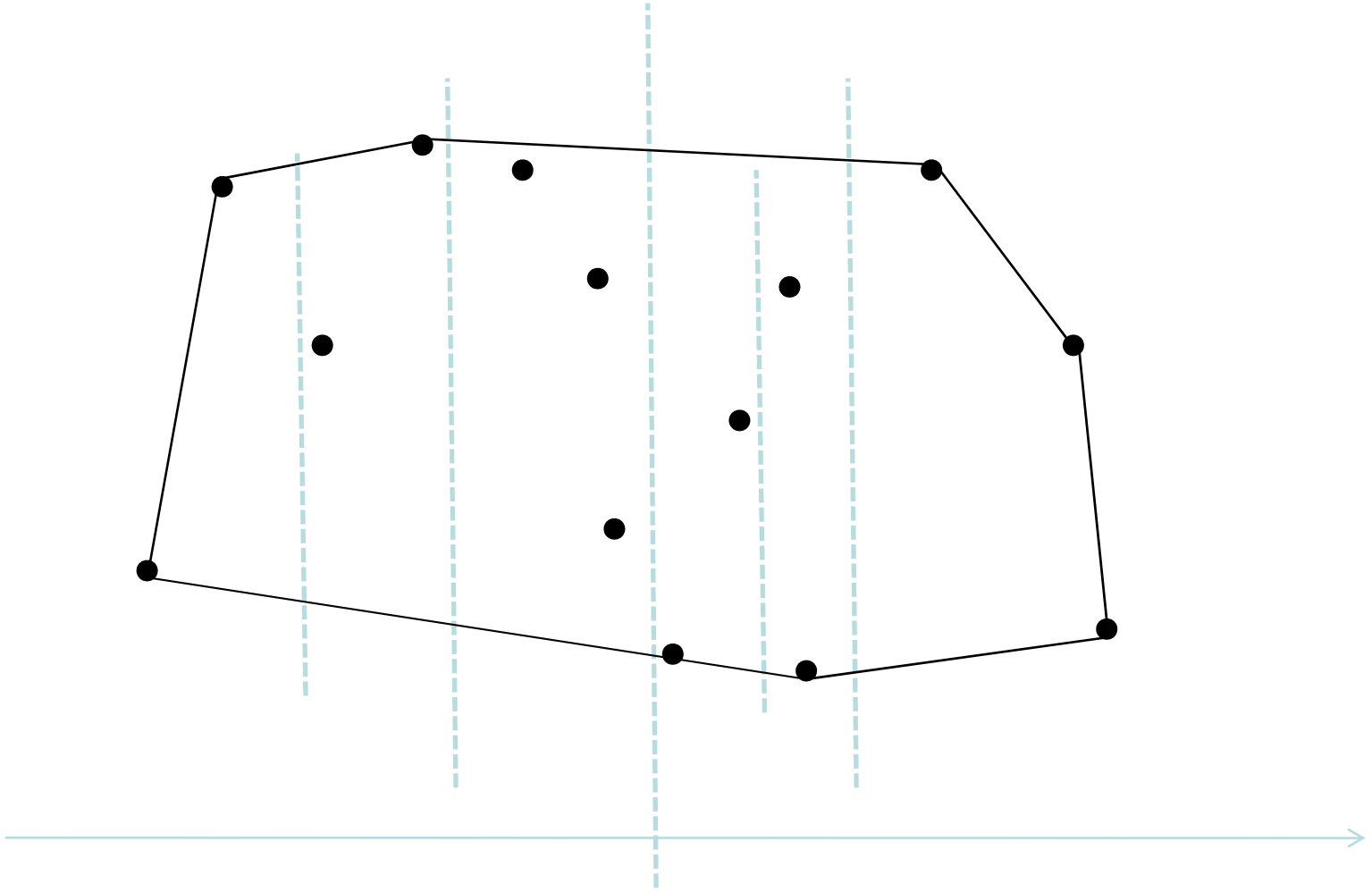
Konveks innhylling



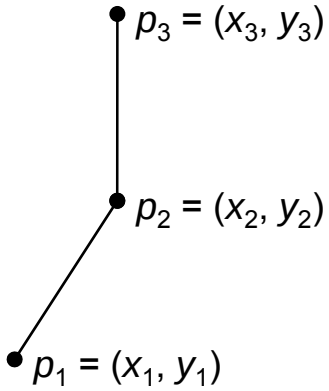
Konveks innhylling



Konveks innhylling

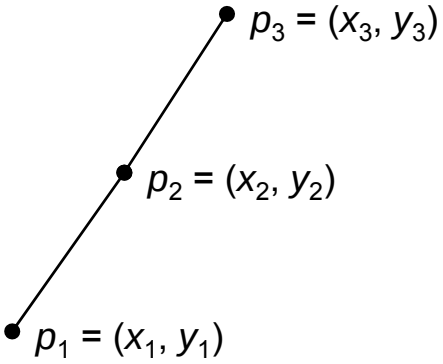


Konveks innhylling



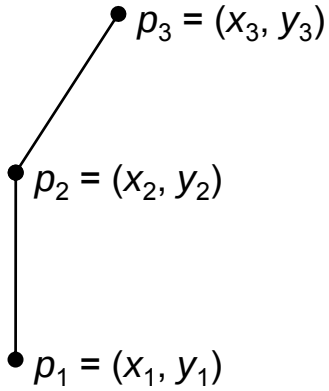
Venstresving

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} > 0$$



Rett fram

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0$$



Høyresving

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} < 0$$

Determinantens fortegn viser om vi gjør en venste- eller høyresving

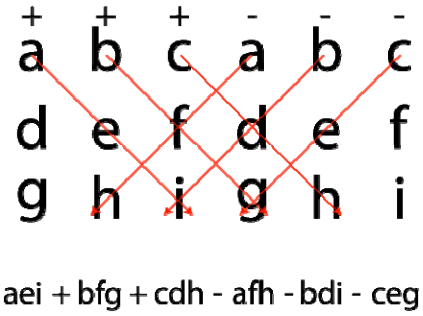
Konveks innhylling

Beregning av determinanter for 3x3-matriser

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

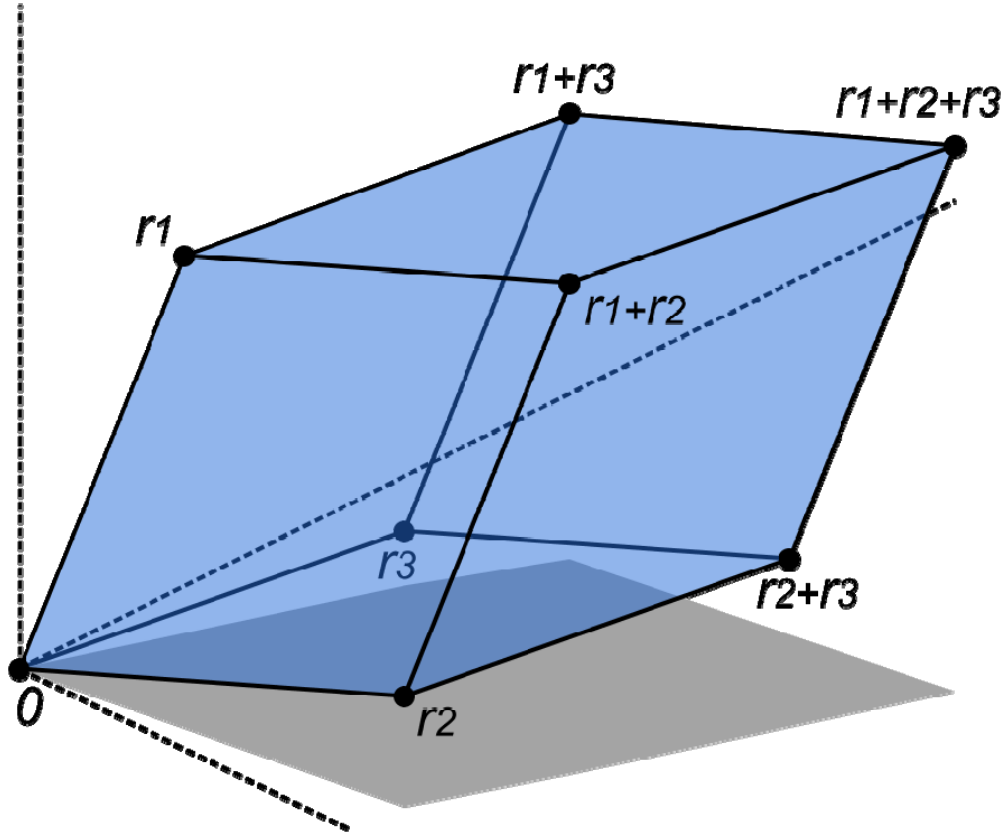
$$\det(A) = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

$$= aei - afh - bdi + bfg + cdh - ceg$$



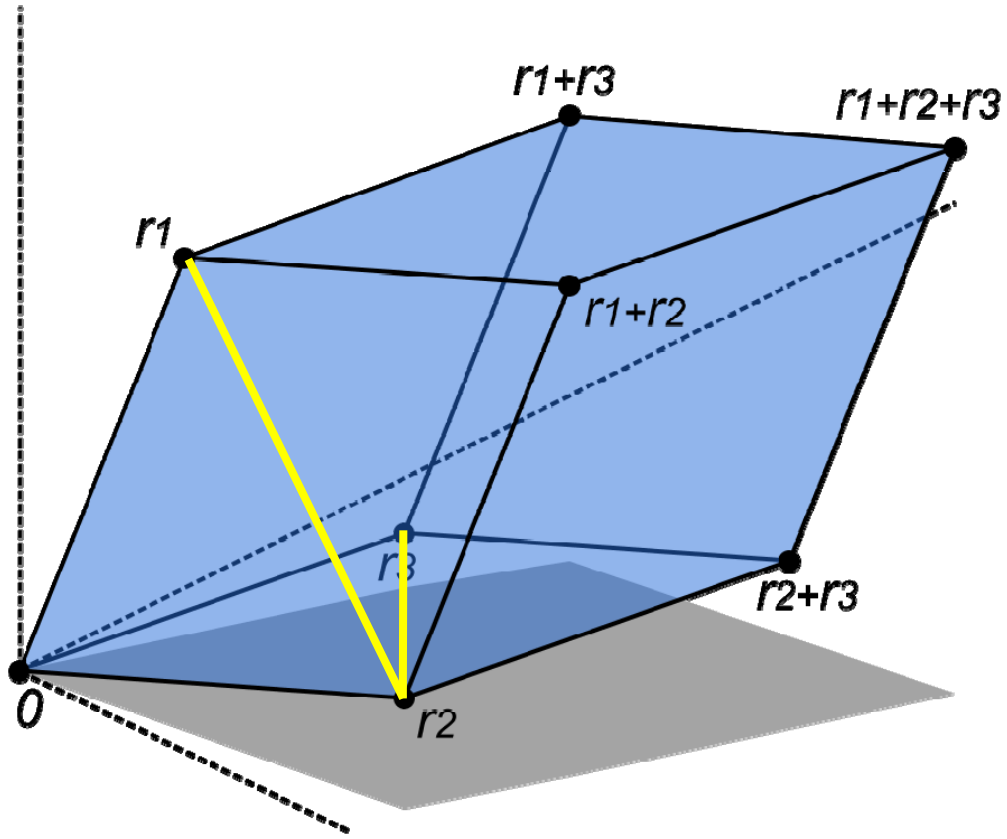
Konveks innhylling

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$



Den geometriske tolkningen til determinanten til matrisen A (det(A), eller bare $|A|$) er volumet til parallellepipedet utspent av rekkevektorene i A. (Eller kolonne-vektorene, det blir det samme).

Konveks innhylling



Konveks innhylling

Vi sorterer nodene etter x-koordinat. Dette tar tid $O(n \log_2 n)$.

Hver gang vi spleiser to innhyllinger må vi potensielt flytte endepunktene på broen(e) $O(n)$ ganger, så hver spleis tar tid $O(n)$

Vi gjør opplagt $\log_2 n$ oppdelinger, siden vi hele tiden deler vår opprinnelig mengde av n noder i to, så det er $\log_2 n$ spleiser.

Total kjøretid $O(n \log_2 n)$.

Denne kjøretiden ($O(n \log_2 n)$ er faktisk optimal, det følger av reduksjonen SORTING $\propto$ CONVEX HULL .

SORTING $\propto$ CONVEX HULL

Gitt *n* heltall skal vi vise at vi kan sortere disse med en algoritme for konveks innhylling.

Som vi husker er $\Omega(n \log_2 n)$ nedre grense for hvor raskt vi kan sortere. (Notat.)

Når vi viser SORTING $\propto$ CONVEX HULL, betyr det at CONVEX HULL er minst like vanskelig som SORTING, slik at om vi kunne løst CONVEX HULL raskere enn $O(n \log_2 n)$, så kunne vi løst SORTING raskere enn $O(n \log_2 n)$ – umulig.

Vi må vise hvordan vi gjør om en SORTING-instans (en mengde tall som skal sorteres) til en CONVEX HULL-instans (punkter vi skal finne innhyllingen av), og hvordan vi skal tolke innhyllingen vi da finner som en sortering av de opprinnelige tallene.

SORTING $\propto$ CONVEX HULL

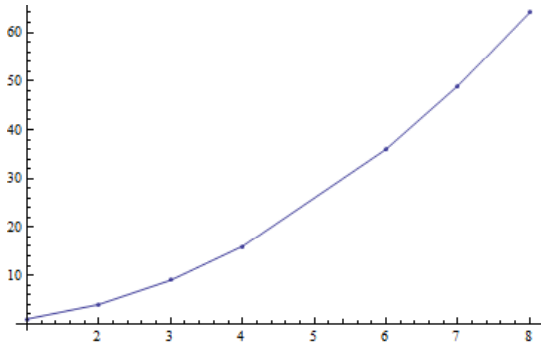
SORTING

T = $<t_1, t_2, \ldots, t_n>$
(usorterte heltall)

$\longrightarrow$

CONVEX HULL

P = $\{(t_1, t_1^2),(t_2, t_2^2),\ldots, (t_n, t_n^2)\}$
(en punktmengde)



Å sortere *n* tall kan altså gjøres med
Følgende rutine som benytter en
CONVEX HULL-algoritme som subrutine:

```
SORTING(T){
      < Lag punktmengden P ut fra T, som beskrevet over >          O(n)
      H = CONVEX_HULL(P)          /* Finner innhyllingen H */      O(n log2 n)
      < Finn nederste punkt i H (lavest y-koordinat) >             O(n)
      < Les av x-koordinaten til punktene i retning mot klokka >   O(n)
}                                                                  O(n log2 n)
```

(Kunne vi løst CONVEX HULL raskere enn $O(n \log_2 n)$, kunne vi også sortert *n* tall raskere enn $O(n \log_2 n)$, som vi vet er umulig.)

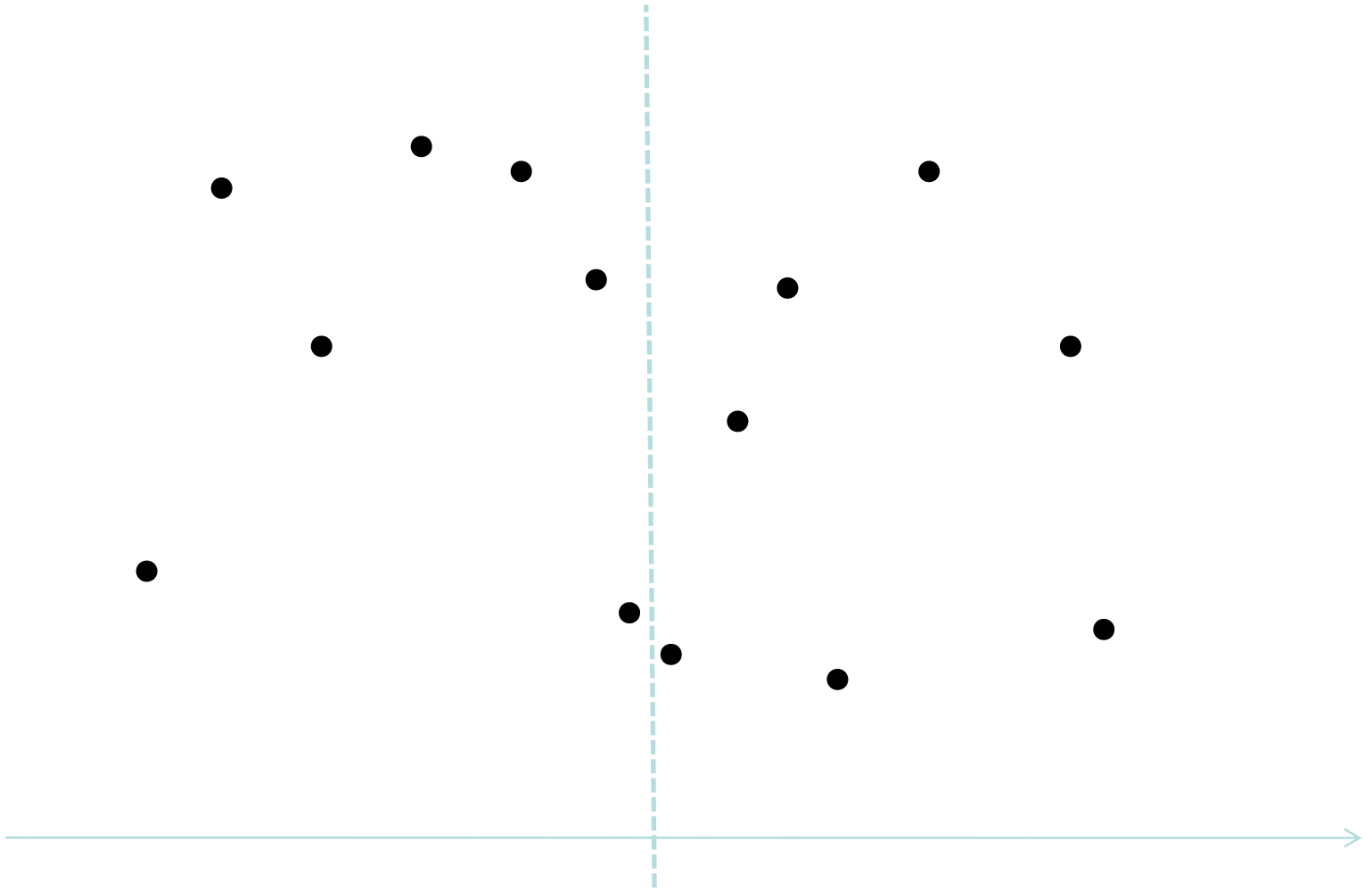
Nærmeste par av punkter

Gitt en mengde punkter i k-dimensjonalt rom, $P \in \mathbf{R}^k$, ønsker vi å finne det par av punkter som har kortest innbyrdes avstand.

(Vi skal igjen, for enkelthets skyld, bare se på $k = 2$.)

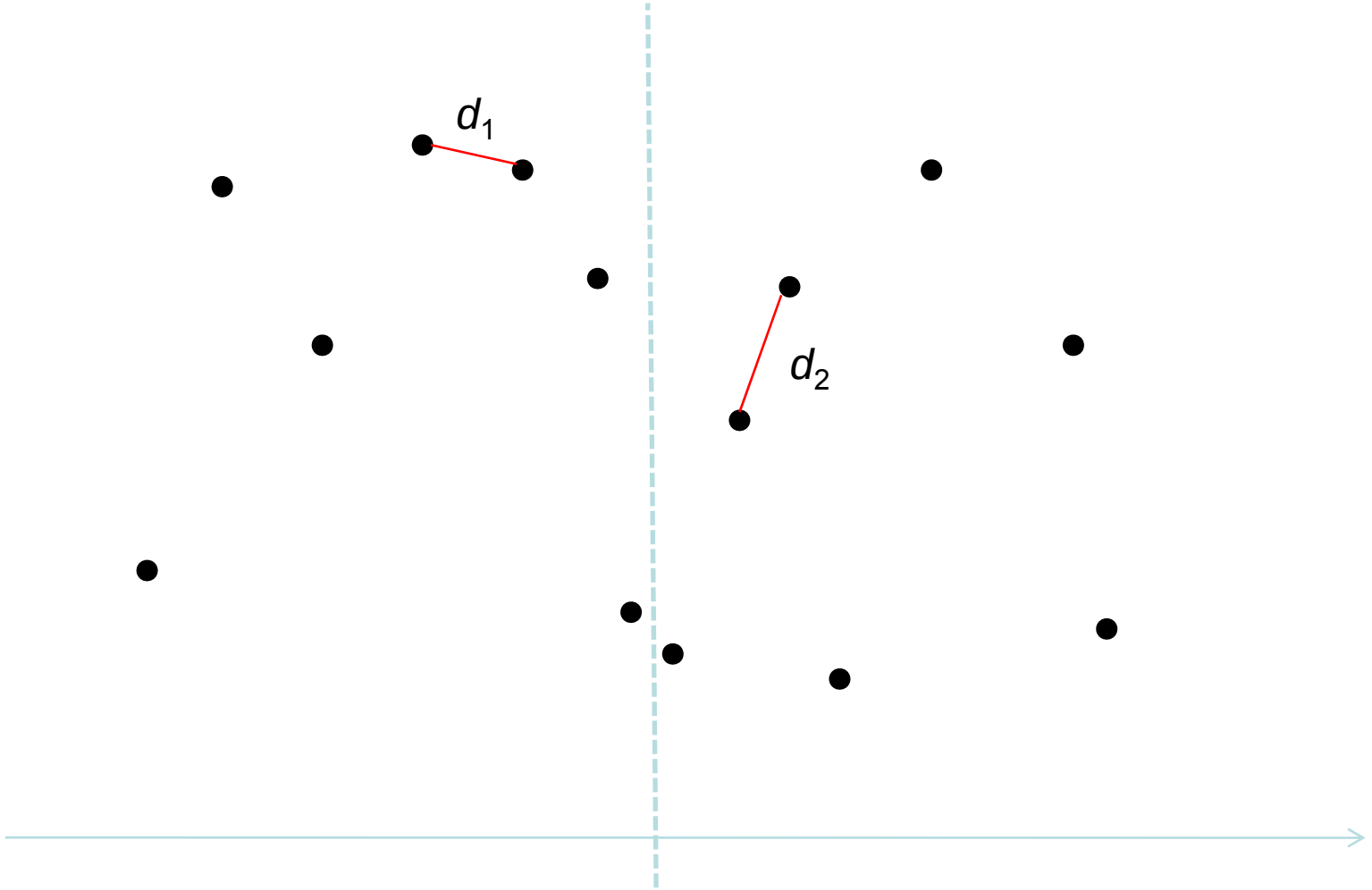
Dette kan vi opplagt gjøre i tid $O(n^2)$ ved å se på alle par, men vi skal lage en mer effektiv algoritme basert på divide-and-conquer.

Nærmeste par av punkter

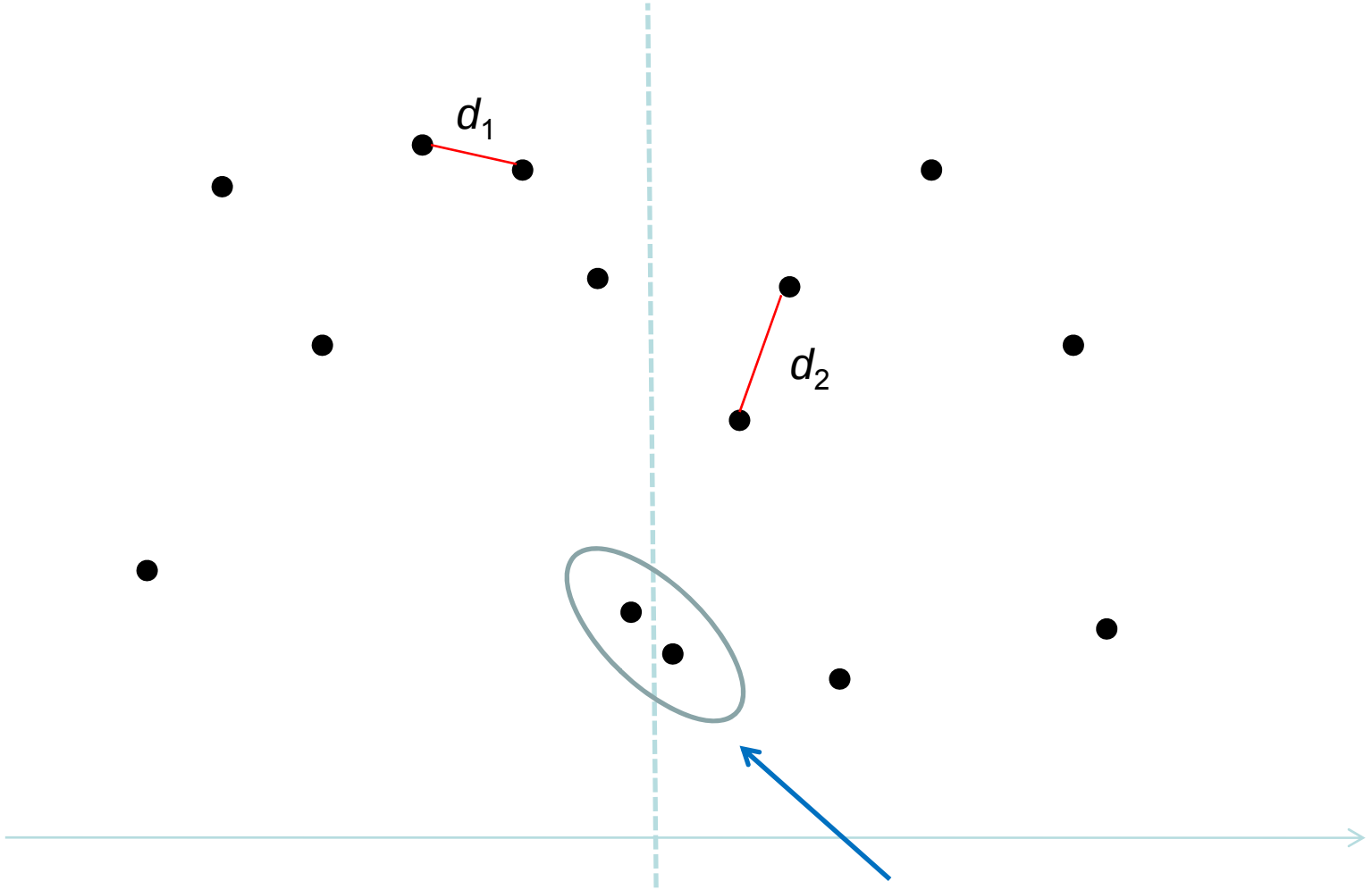


Deler i to ved *x*-median, helt til vi får 2 eller 3 punkter i mengden, og finner korteste avstand.

Nærmeste par av punkter

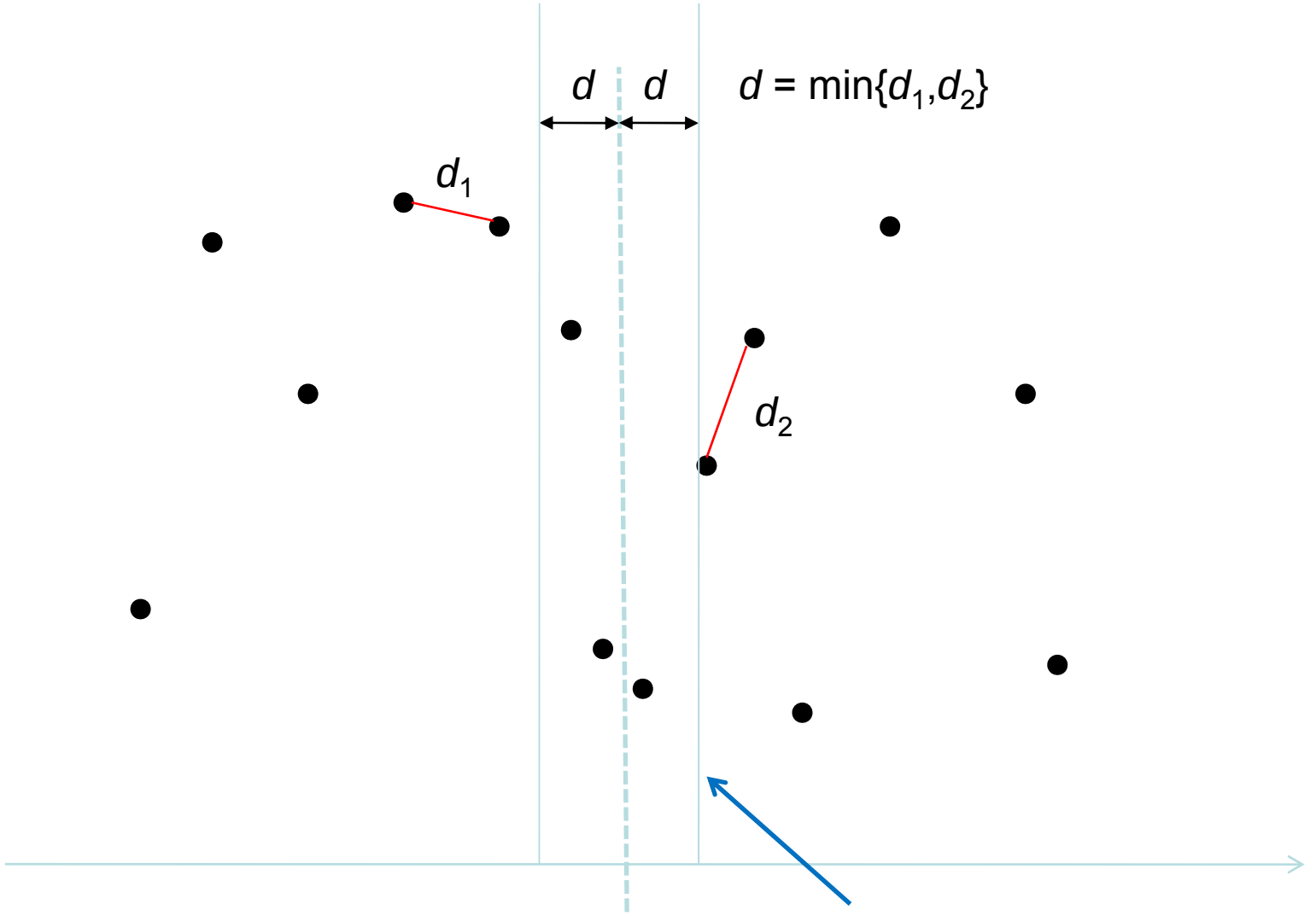


Nærmeste par av punkter

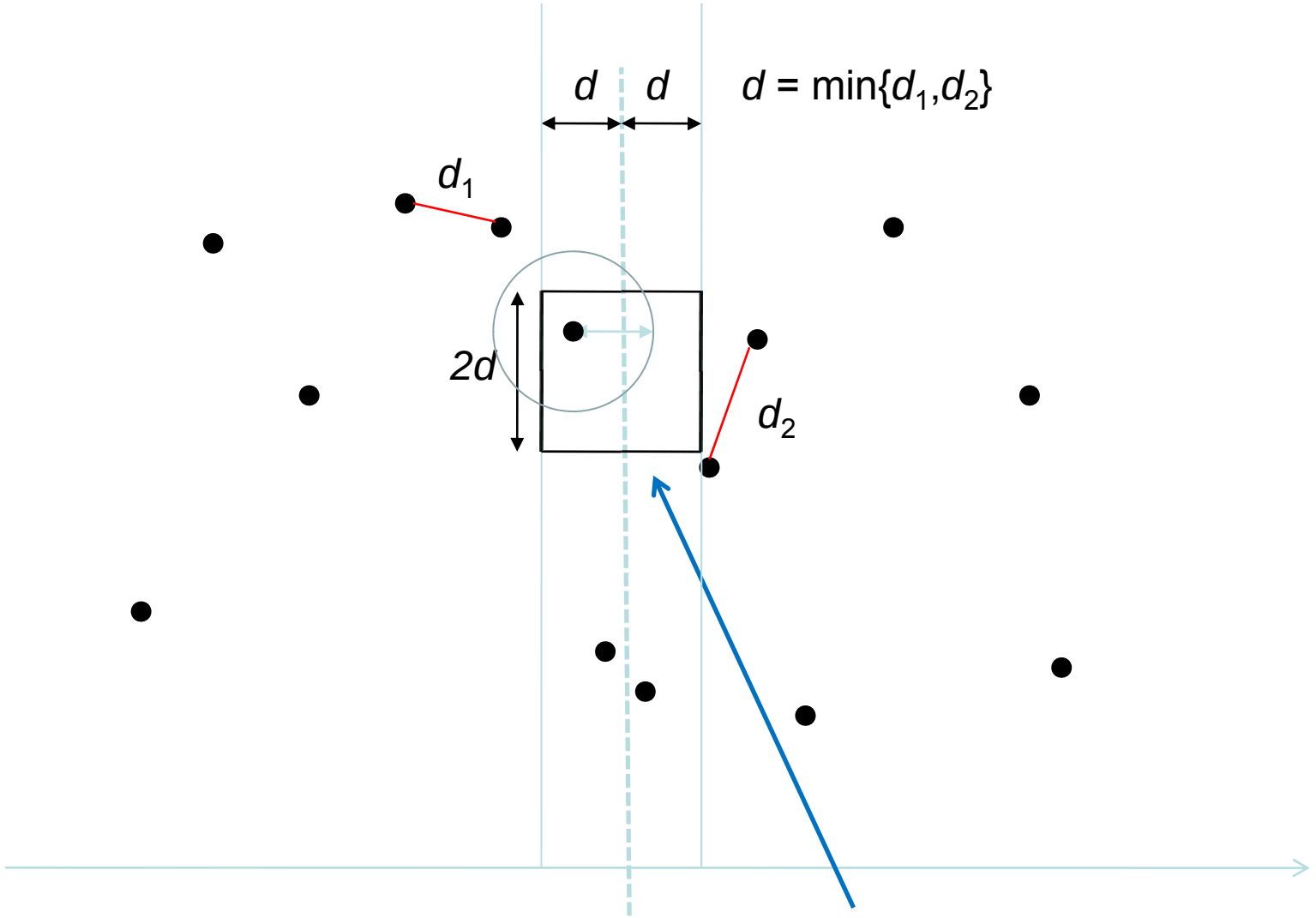


Ikke sikkert korteste avstand fra en av mengdene er kortest i den spleisede mengden.

Nærmeste par av punkter

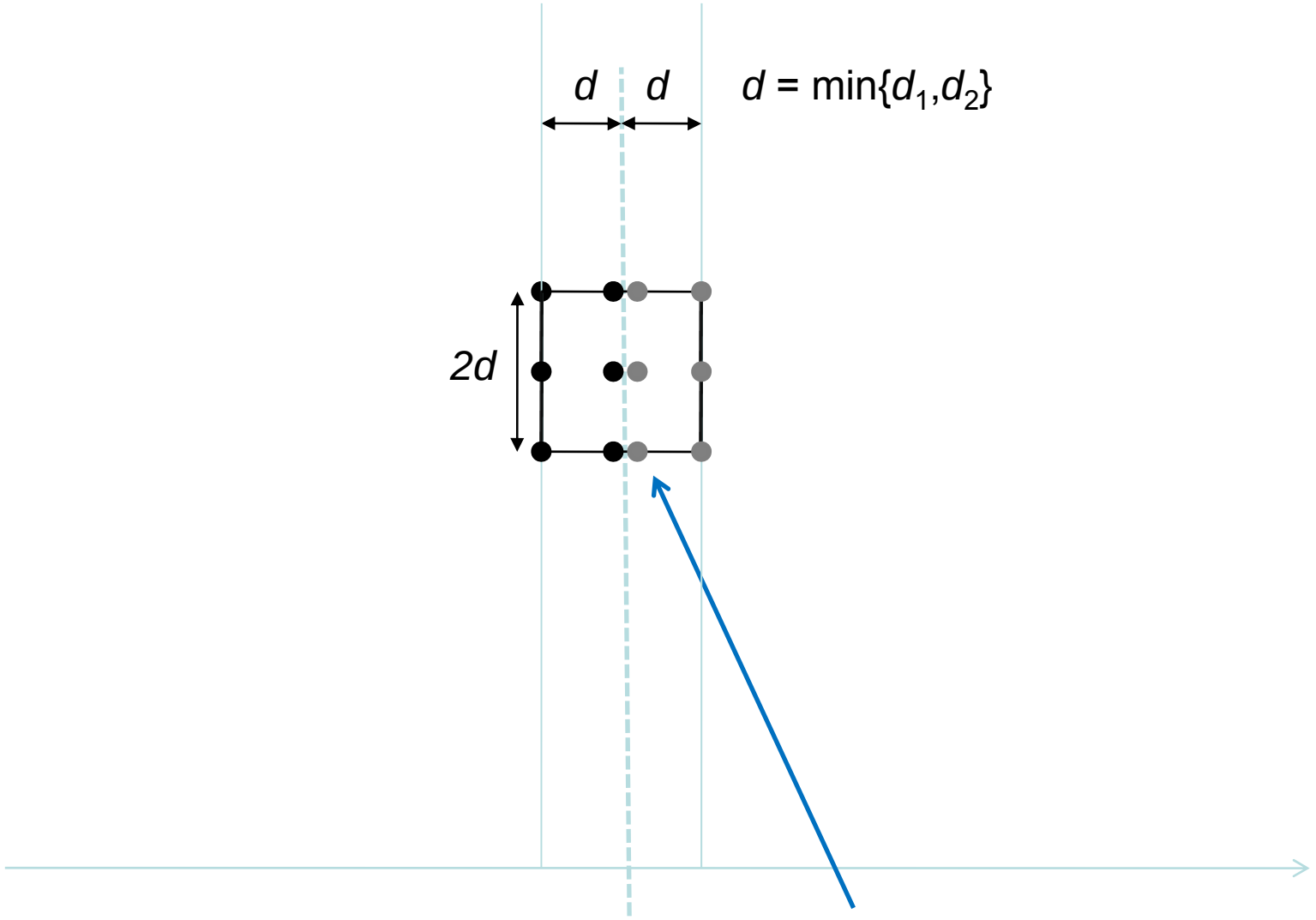


Nærmeste par av punkter

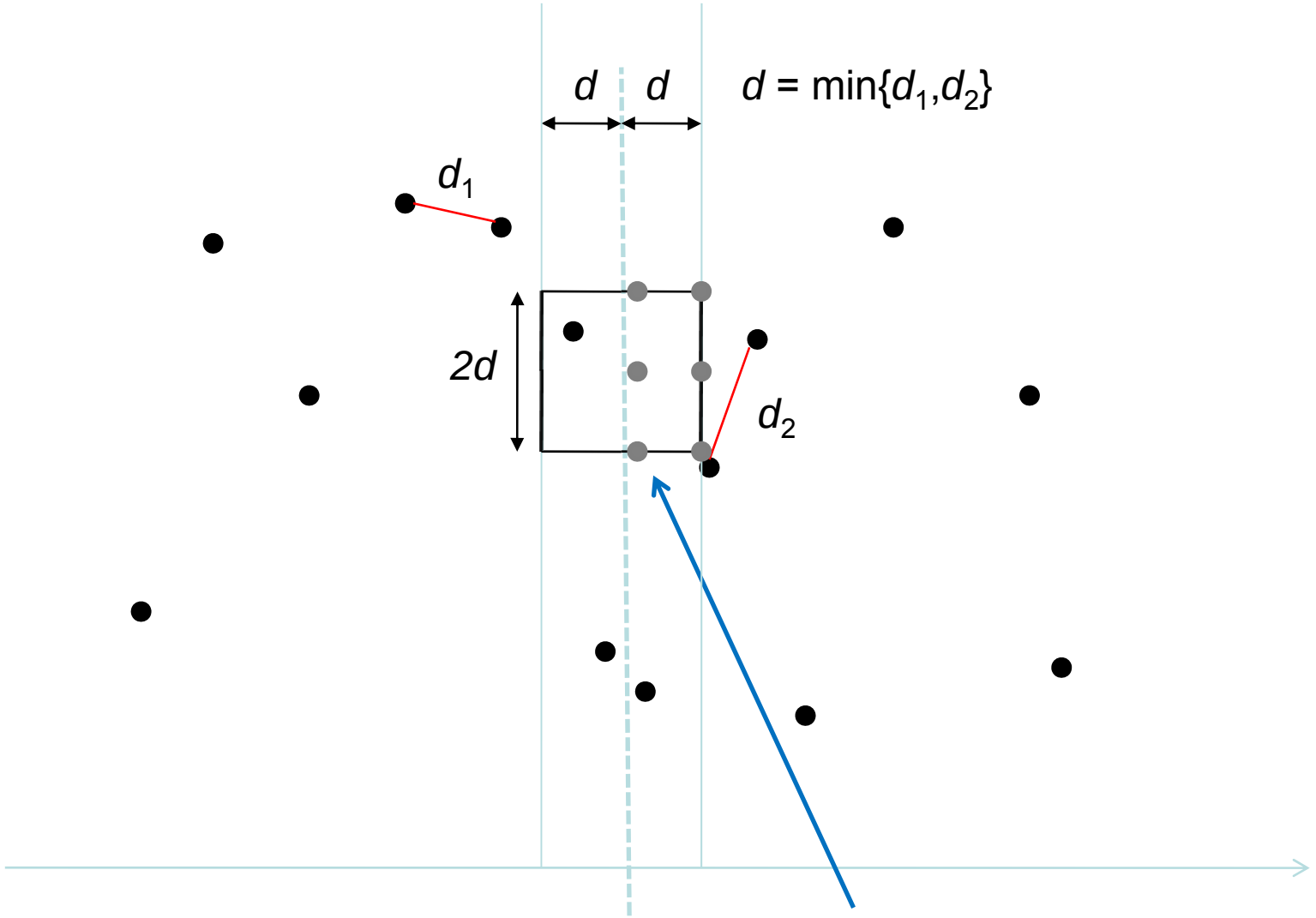


I en slik firkant (2*d* x 2*d* stor), kan det ikke ligge punkter med innbyrdes avstand mindre enn *d* (*d* er jo minimum).

Nærmeste par av punkter



Nærmeste par av punkter



For hver node på venstre side er det altså maksimalt 6 noder på høyre side som må sjekkes.

Nærmeste par av punkter

Vi sorterer nodene både etter x- og y-koordinat. Dette tar tid $O(n \log_2 n)$.

Spleise-steget tar tid $O(n)$ (for hver node i den ene siden av ”skjøten” er det bare et konstant (6) antall noder som må sjekkes på den andre siden. Ettersom nodene er sortert på både x- og y-koordinat er det greit å holde rede på hvilke noder det er snakk om.

Vi gjør opplagt $\log_2 n$ oppdelinger, siden vi hele tiden deler vår opprinnelig mengde av n noder i to, så det er $\log_2 n$ spleiser.

Total kjøretid $O(n \log_2 n)$.