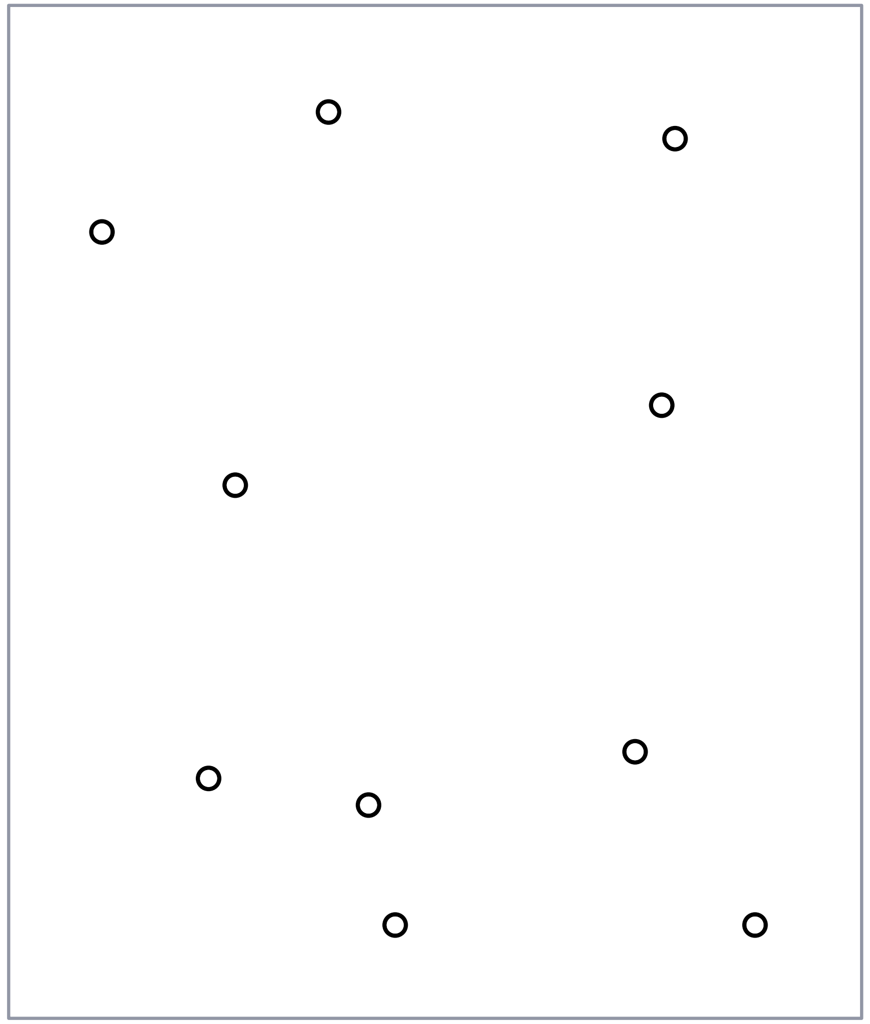
Well-Separated Pair Decomposition

Application: geometric spanners

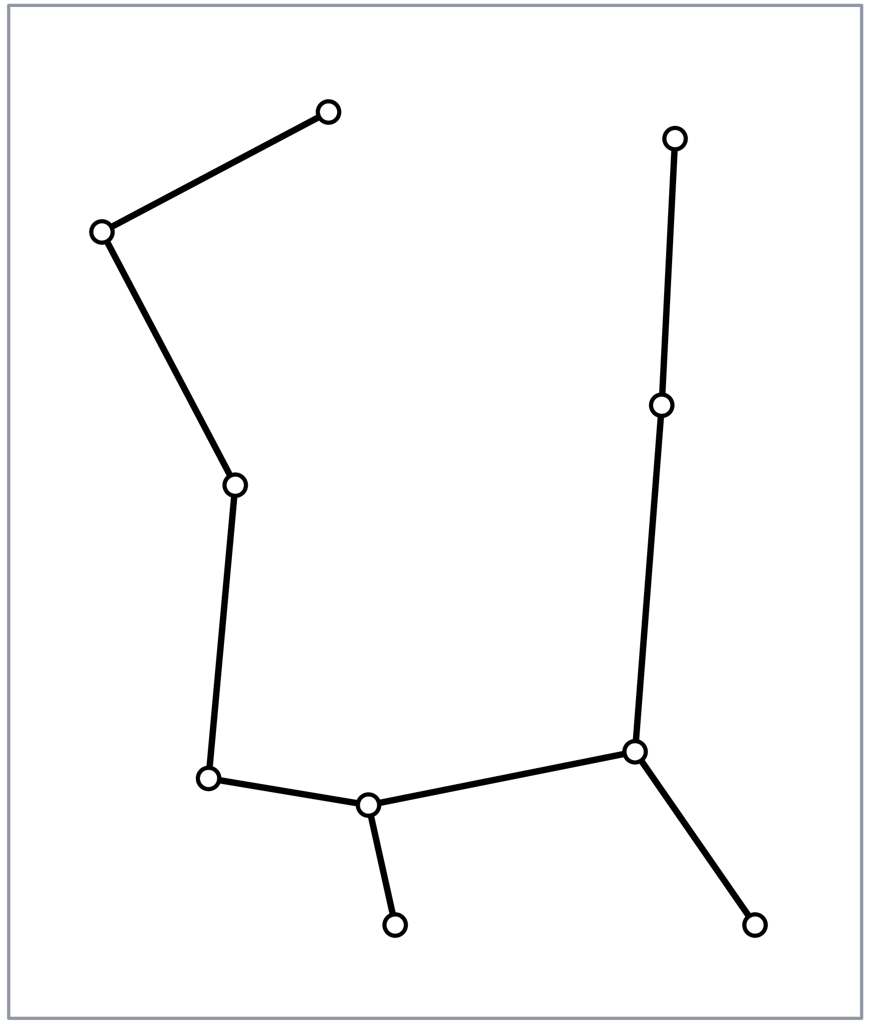
Construction and size

Motivation



Problem: Connect a set of cities by a new street network.

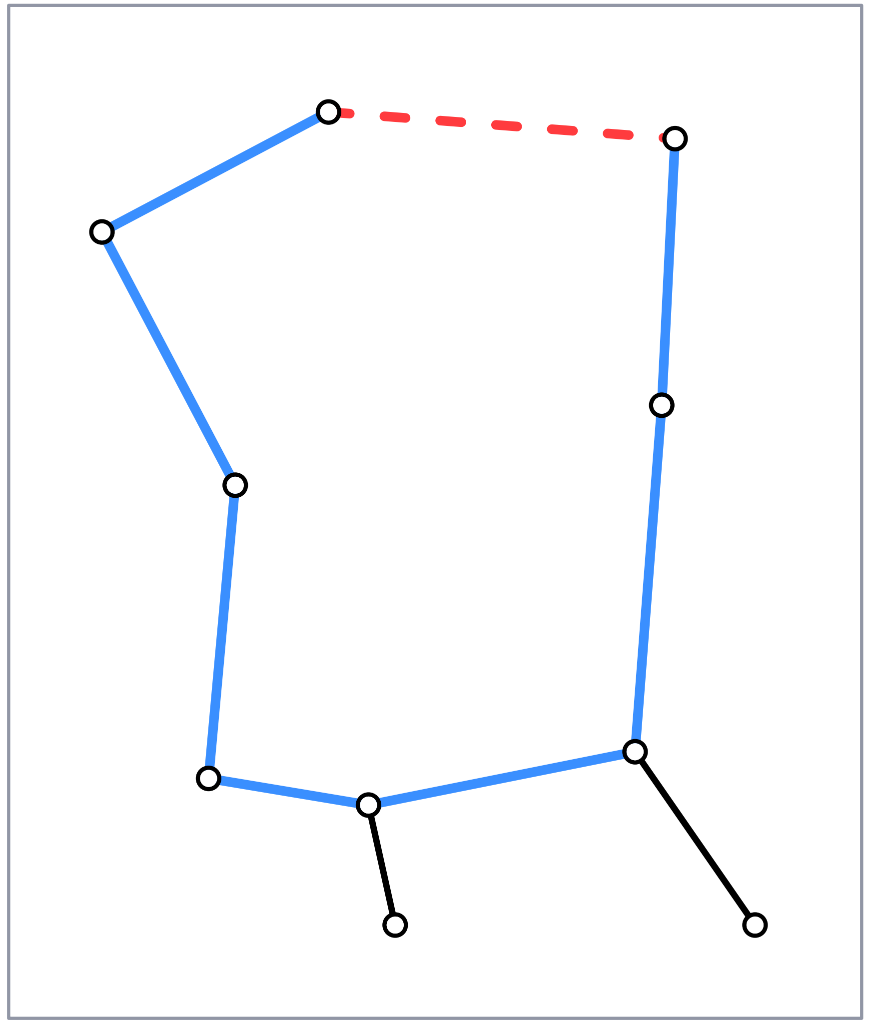
Motivation



Problem: Connect a set of cities by a new street network.

1. Idea: Euclidean MST

Motivation

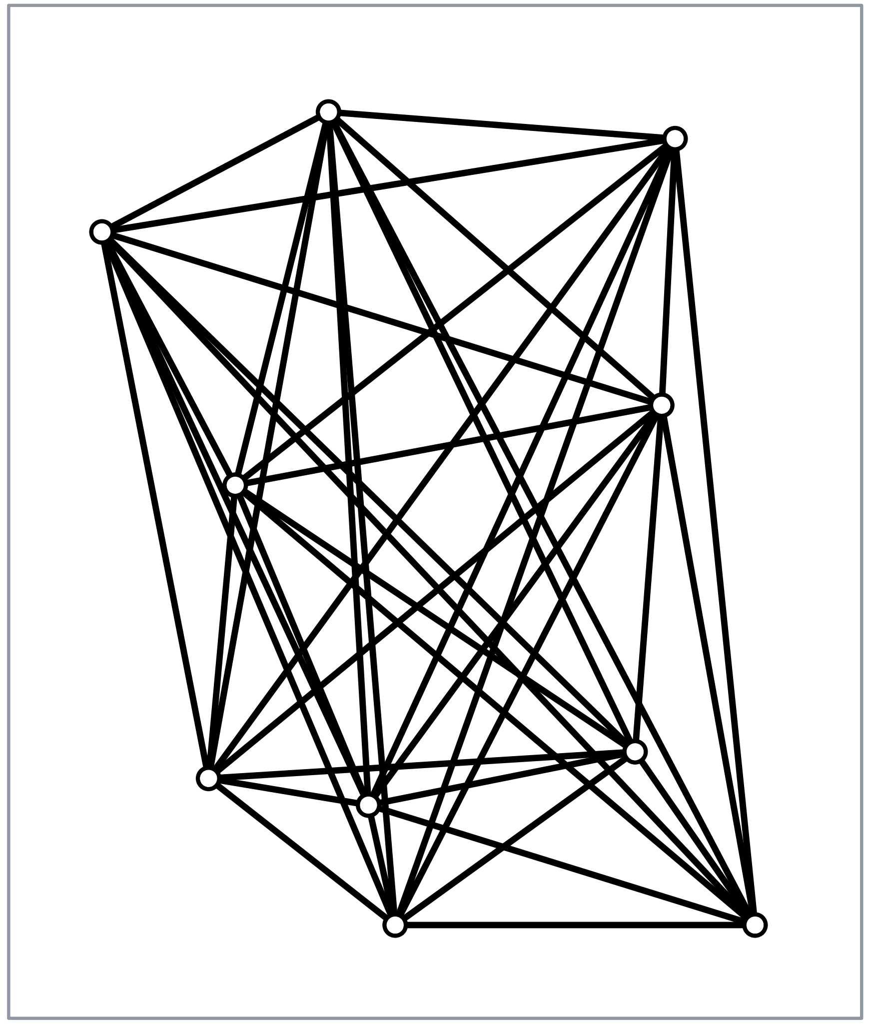


Problem: Connect a set of cities by a new street network.

1. Idea: Euclidean MST

But for any pair (x, y) the graph distance shouldn't be much longer than $\|x - y\|$

Motivation



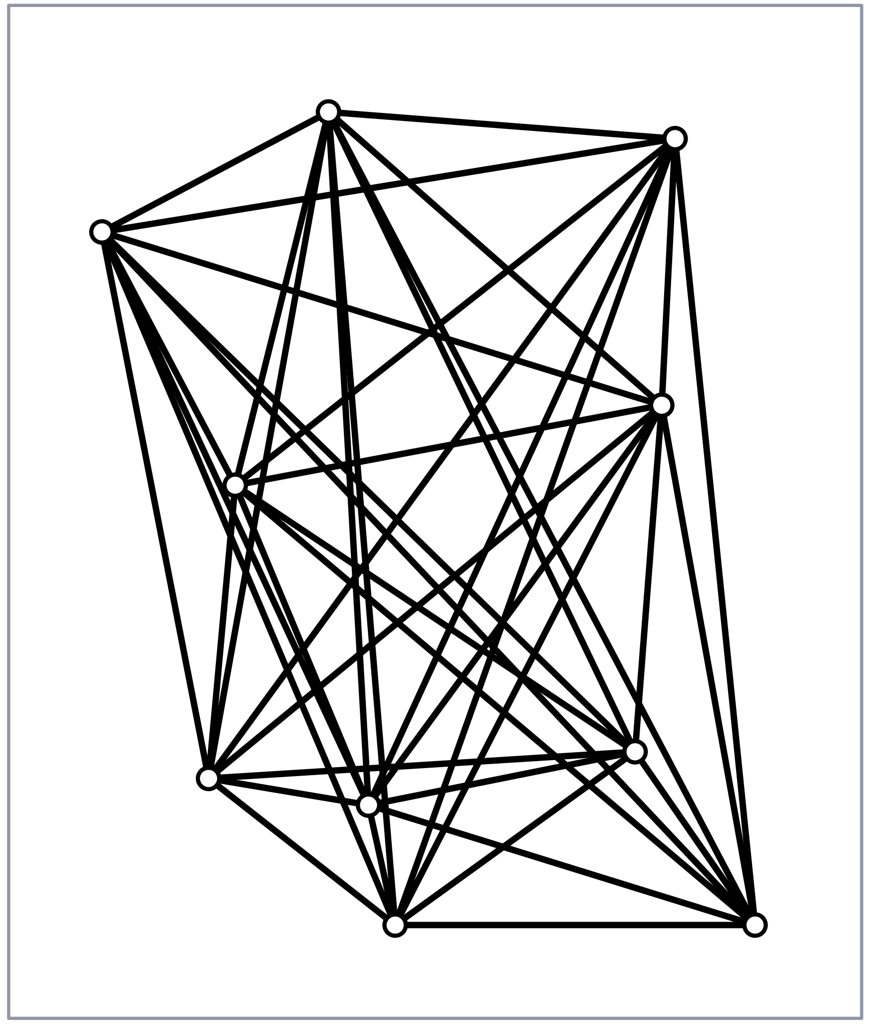
Problem: Connect a set of cities by a new street network.

1. Idea: Euclidean MST

   But for any pair (x, y) the graph distance shouldn't be much longer than $\|x - y\|$

2. Idea: complete graph

Motivation



Problem: Connect a set of cities by a new street network.

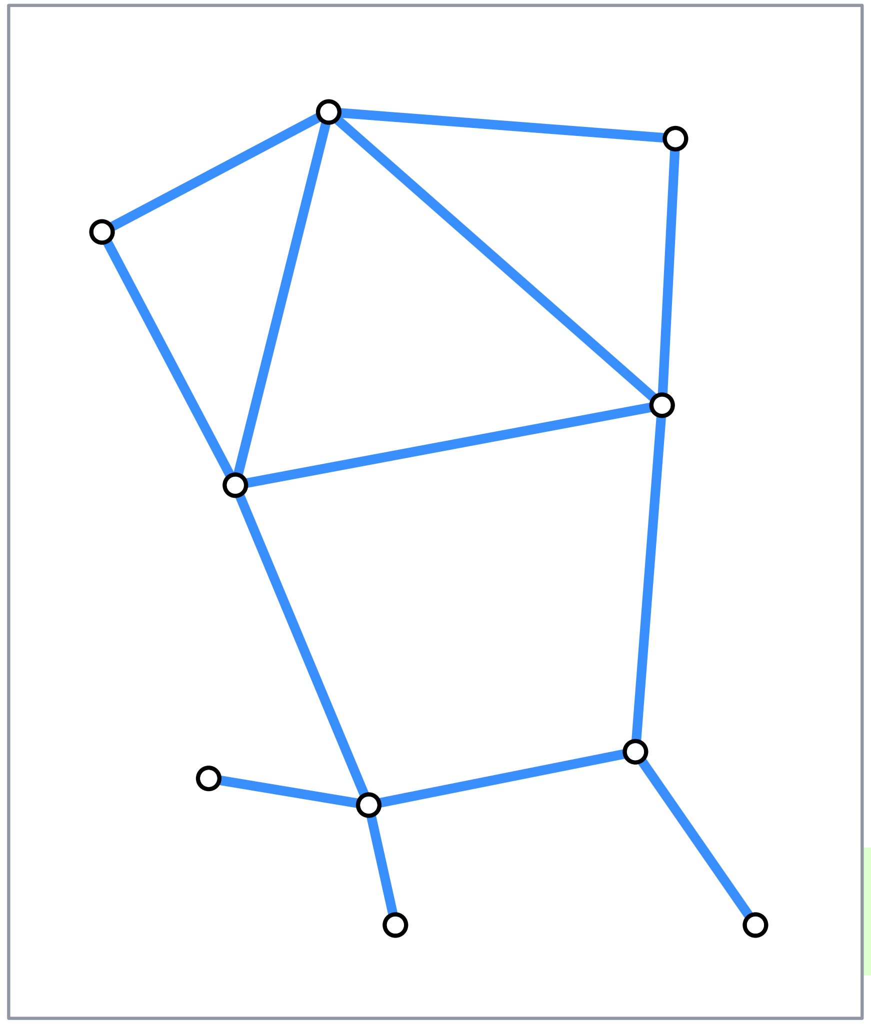
1. Idea: Euclidean MST

But for any pair (x, y) the graph distance shouldn't be much longer than $\|x - y\|$

2. Idea: complete graph

The budget for roads only pays for $O(n)$ roads.

Motivation



Problem: Connect a set of cities by a new street network.

1. Idea: Euclidean MST

   But for any pair (x, y) the graph distance shouldn't be much longer than $\|x - y\|$

2. Idea: complete graph

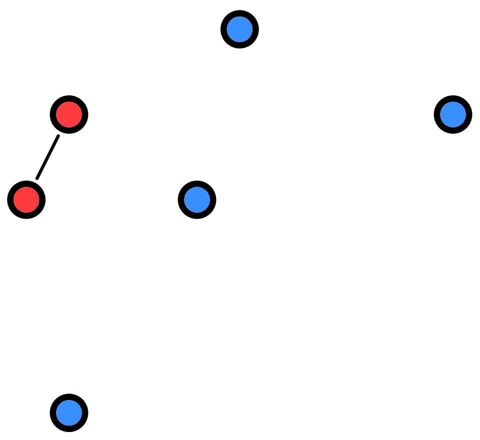
   The budget for roads only pays for $O(n)$ roads.

3. Idea: sparse t-spanner

$O(n)$ edges

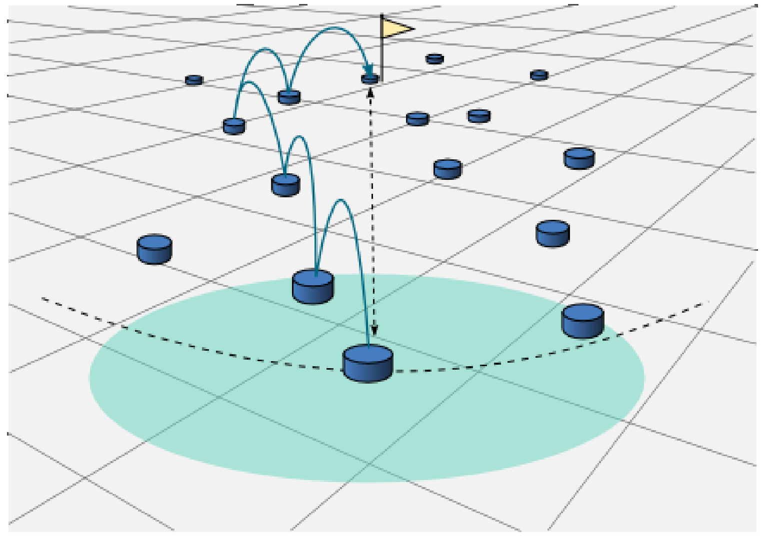
detour $\leq t \cdot$ shortest path

Applications of distance approximation



fast, approximate distance computation

- geometric approximation algorithms for diameter, minimum spanning tree etc.
- exact algorithms: closest pair, nearest neighbor graph, Voronoi diagrams etc.



communication and connectivity in networks

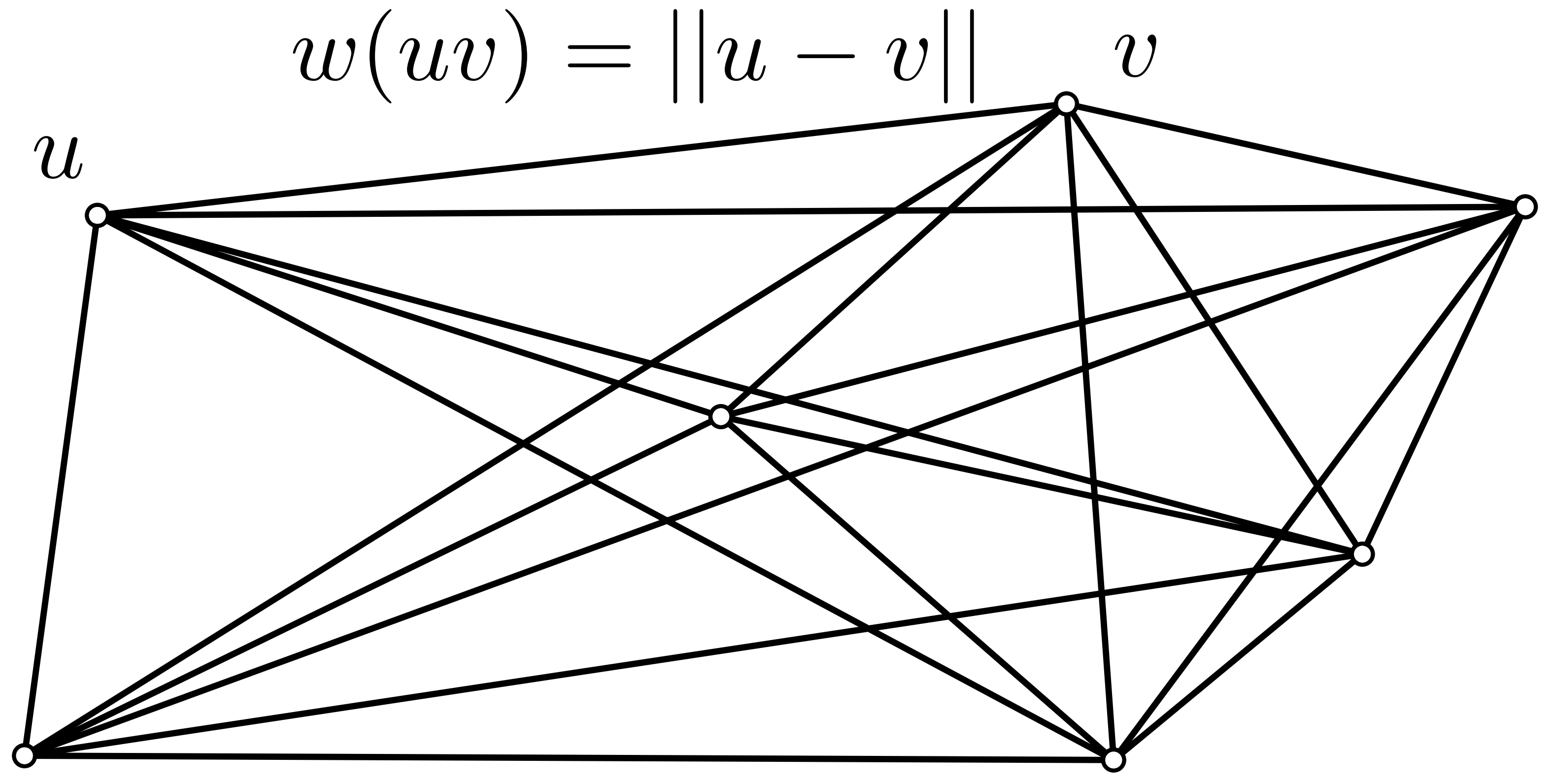
- topology control in wireless networks
- routing in networks
- network analysis

t-spanner

For a set P of n points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{EG}(P) = (P, \binom{P}{2})$ is the complete, weighted graph with Euclidean distances as edge weights.

t-spanner

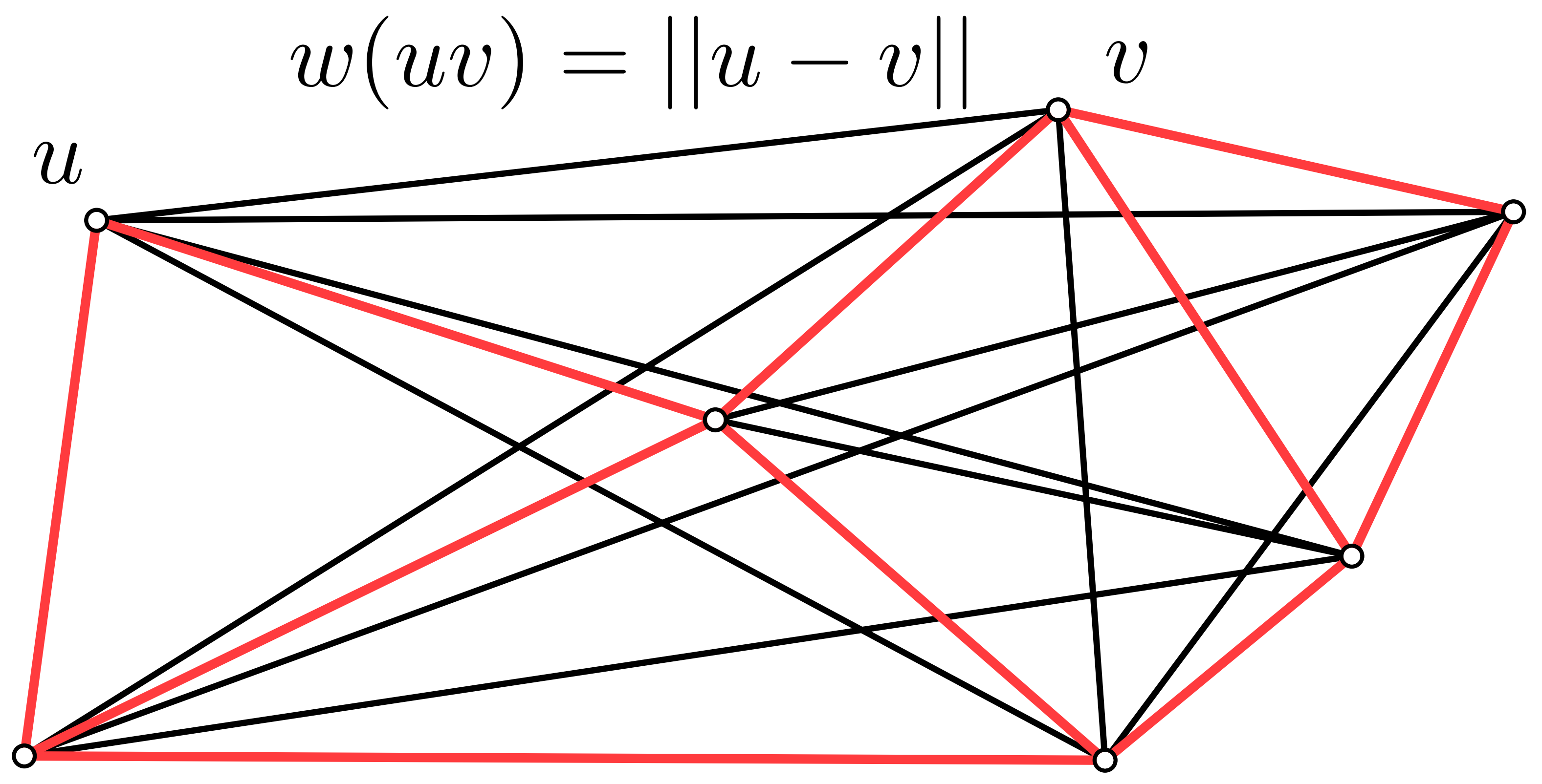
For a set P of n points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{EG}(P) = (P, \binom{P}{2})$ is the complete, weighted graph with Euclidean distances as edge weights.



t-spanner

For a set P of n points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{EG}(P) = (P, \binom{P}{2})$ is the complete, weighted graph with Euclidean distances as edge weights.

Since $\mathcal{EG}(P)$ has $\Theta(n^2)$ edges, we want a sparse graph with $O(n)$ edges such that the shortest paths in the graph approximate the edge weights of $\mathcal{EG}(P)$.



t-spanner

For a set P of n points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{EG}(P) = (P, \binom{P}{2})$ is the complete, weighted graph with Euclidean distances as edge weights.

Since $\mathcal{EG}(P)$ has $\Theta(n^2)$ edges, we want a sparse graph with $O(n)$ edges such that the shortest paths in the graph approximate the edge weights of $\mathcal{EG}(P)$.

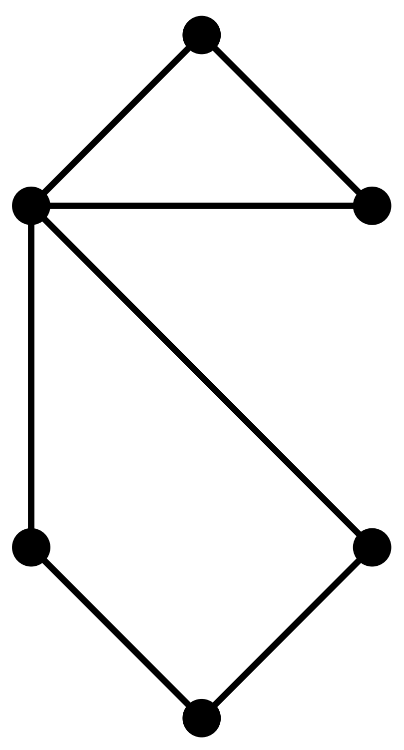
Definition: A weighted graph G with vertex set P is called t-spanner for P and a stretch factor $t \geq 1$, if for all pairs $x, y \in P$:

$$||xy|| \leq \delta_G(x, y) \leq t \cdot ||xy|| \, ,$$

where $\delta_G(x, y) =$ length of the shortest x-to-y path in G.

Quiz

What is the smallest t for which the following graph is a t-spanner?



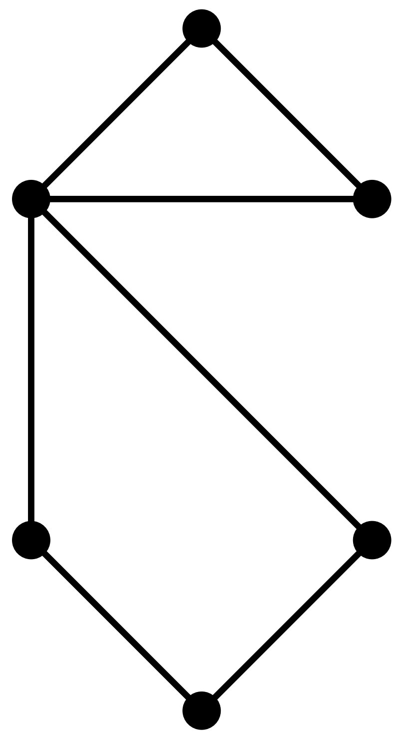
A: $\sqrt{2}$

B: 2

C: $\sqrt{2}+1$

Quiz

What is the smallest t for which the following graph is a t-spanner?



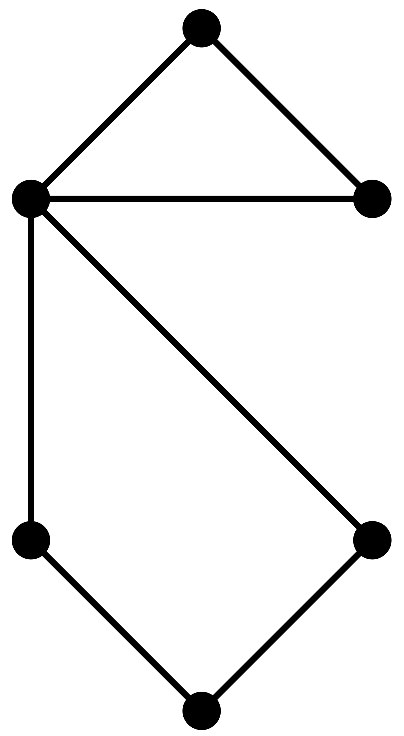
A: $\sqrt{2}$

B: 2

$\boxed{\text{C: } \sqrt{2}+1}$

Quiz

What is the smallest t for which the following graph is a t-spanner?



A: $\sqrt{2}$

B: 2

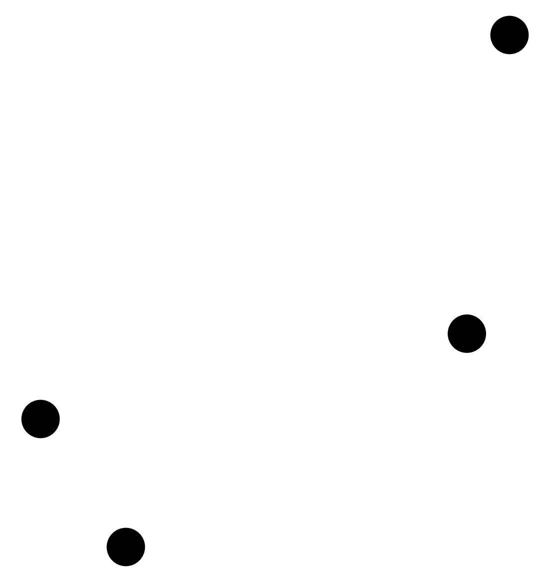
$\boxed{\text{C: } \sqrt{2}+1}$

How can we compute a t-spanner?

Spanner construction paradigms

greedy

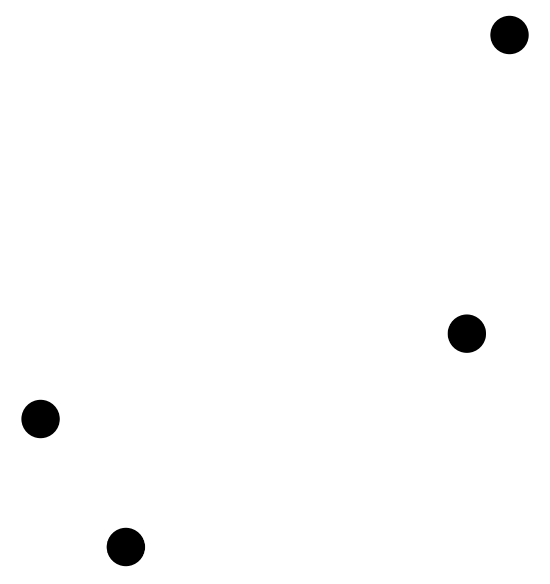
- sort point pairs by distance, start with no edges
- if for the next point pair the dilation is $> t$ then add corresponding edge



Spanner construction paradigms

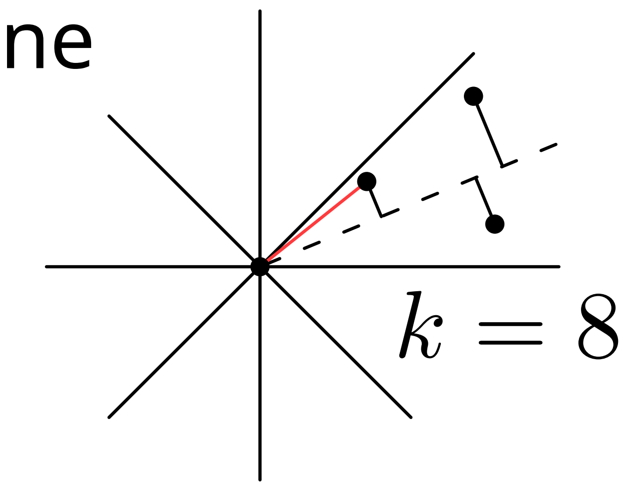
greedy

- sort point pairs by distance, start with no edges
- if for the next point pair the dilation is $> t$ then add corresponding edge



cone-based

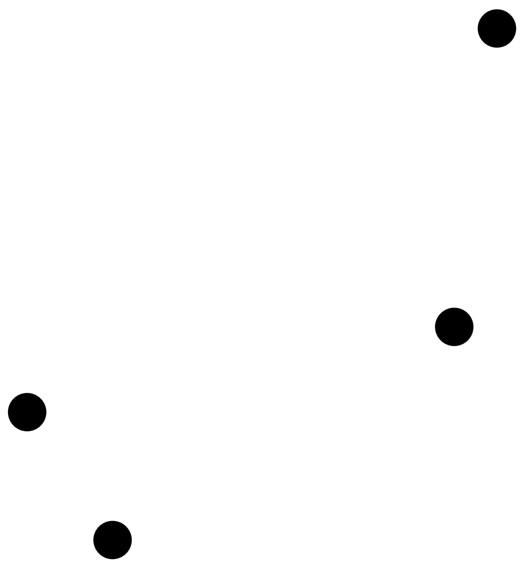
- subdivide space around each point into $k > 6$ non-overlapping cones with angle $\phi = 2\pi/k$
- connect to “closest” point in each cone



Spanner construction paradigms

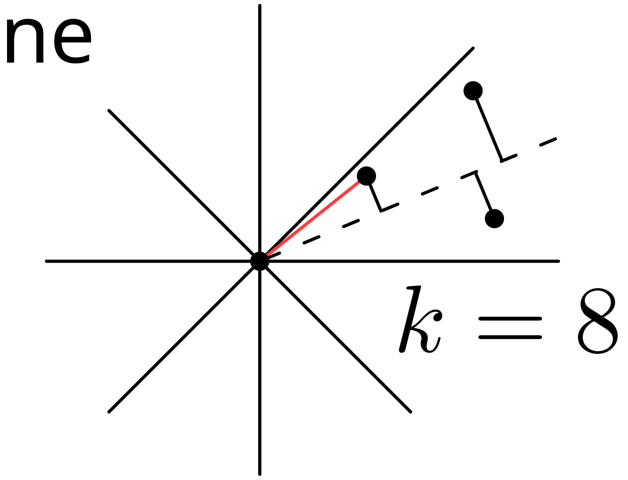
greedy

- sort point pairs by distance, start with no edges
- if for the next point pair the dilation is $> t$ then add corresponding edge



cone-based

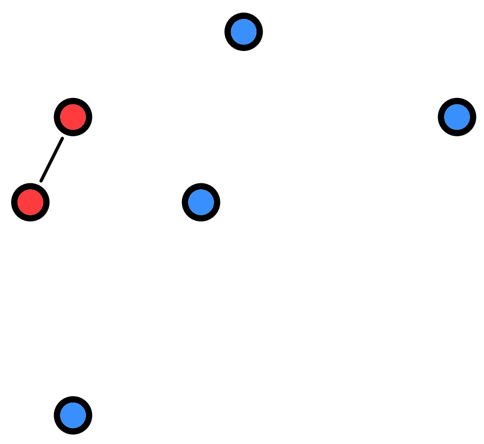
- subdivide space around each point into $k > 6$ non-overlapping cones with angle $\phi = 2\pi / k$
- connect to “closest” point in each cone



distance approximation

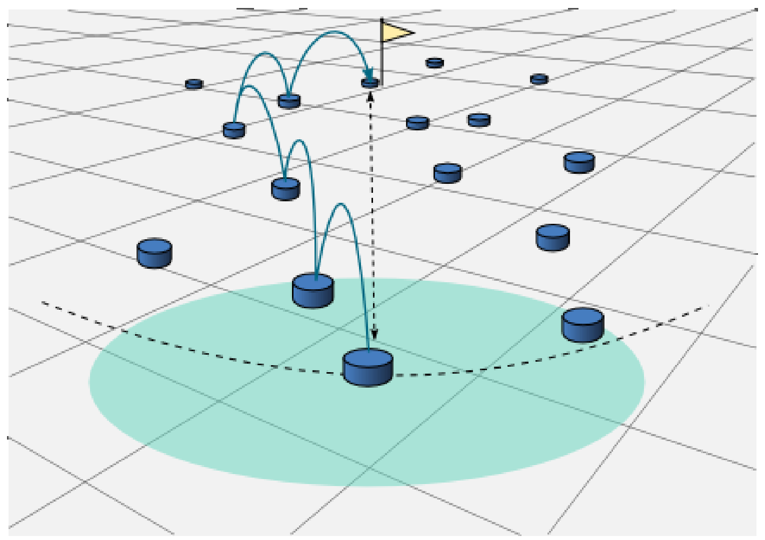
- well-separated pair decomposition (next!)

Applications of distance approximation



fast, approximate distance computation

- geometric approximation algorithms for diameter, minimum spanning tree etc.
- exact algorithms: closest pair, nearest neighbor graph, Voronoi diagrams etc.



communication and connectivity in networks

- topology control in wireless networks
- routing in networks
- network analysis

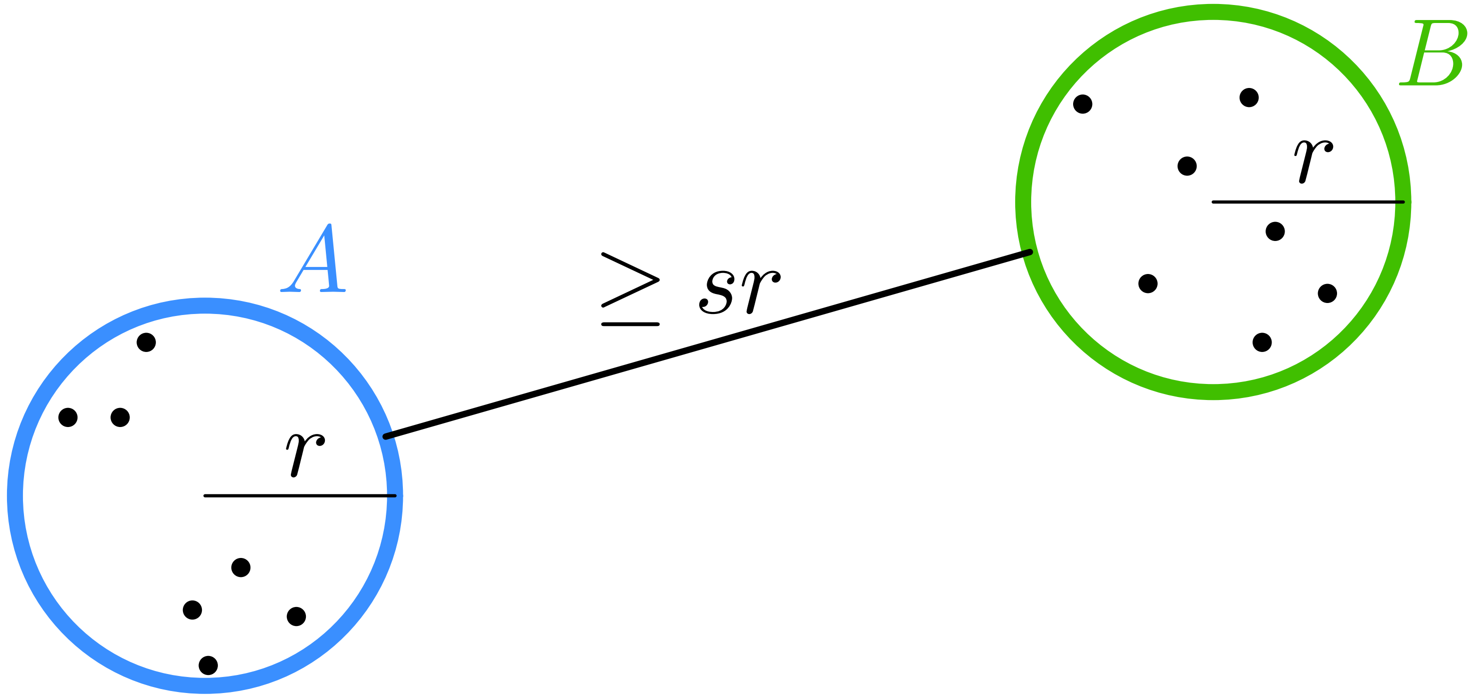
Well-Separated Pair Decomposition

Definition

Reminder: Compressed Quadtrees

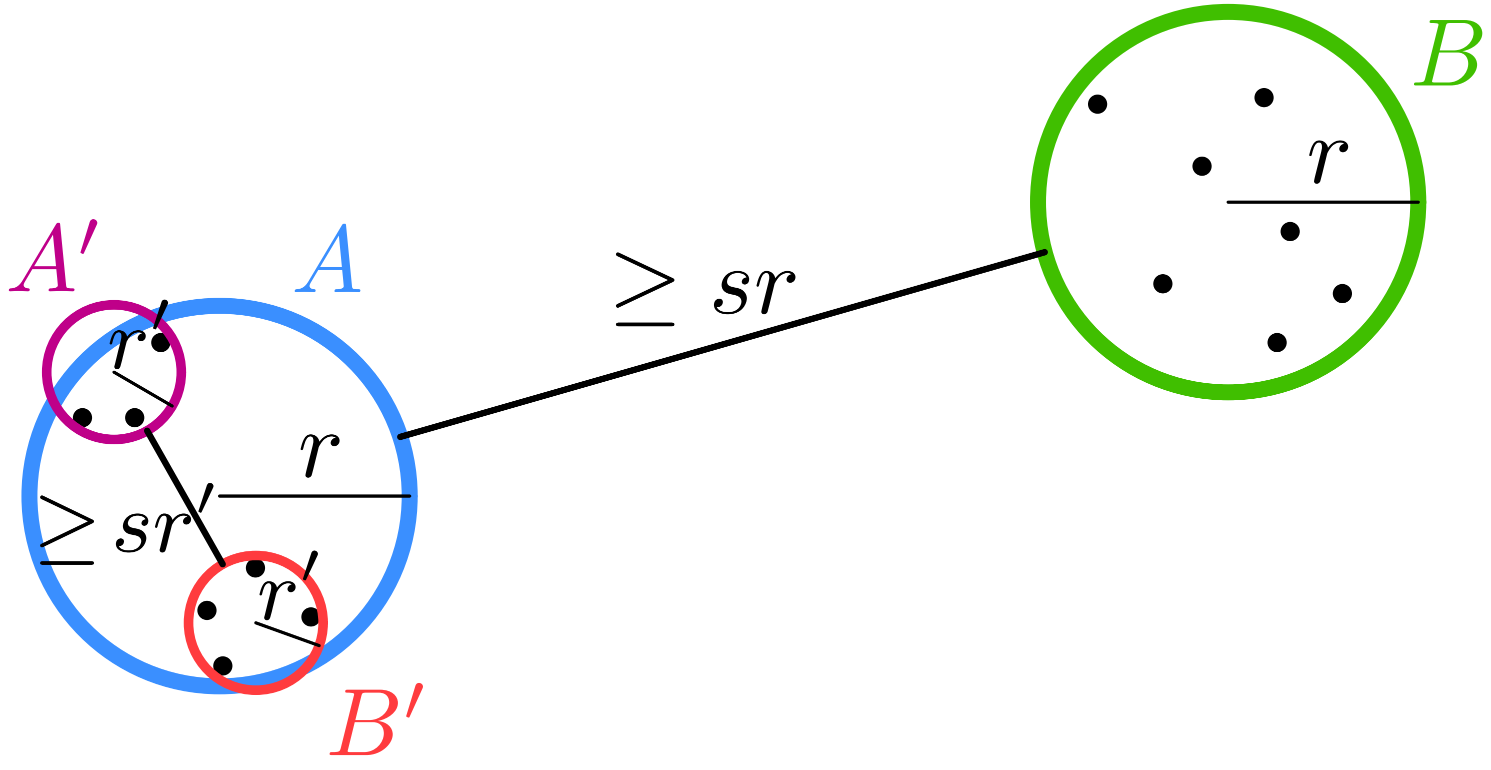
Well-Separated Pairs

Definition: A pair of disjoint point sets A and B in $\mathbb{R}^d$ is called s-well separated for an $s > 0$, if A and B both can be covered by a ball of radius r and the distance between the balls is at least sr.



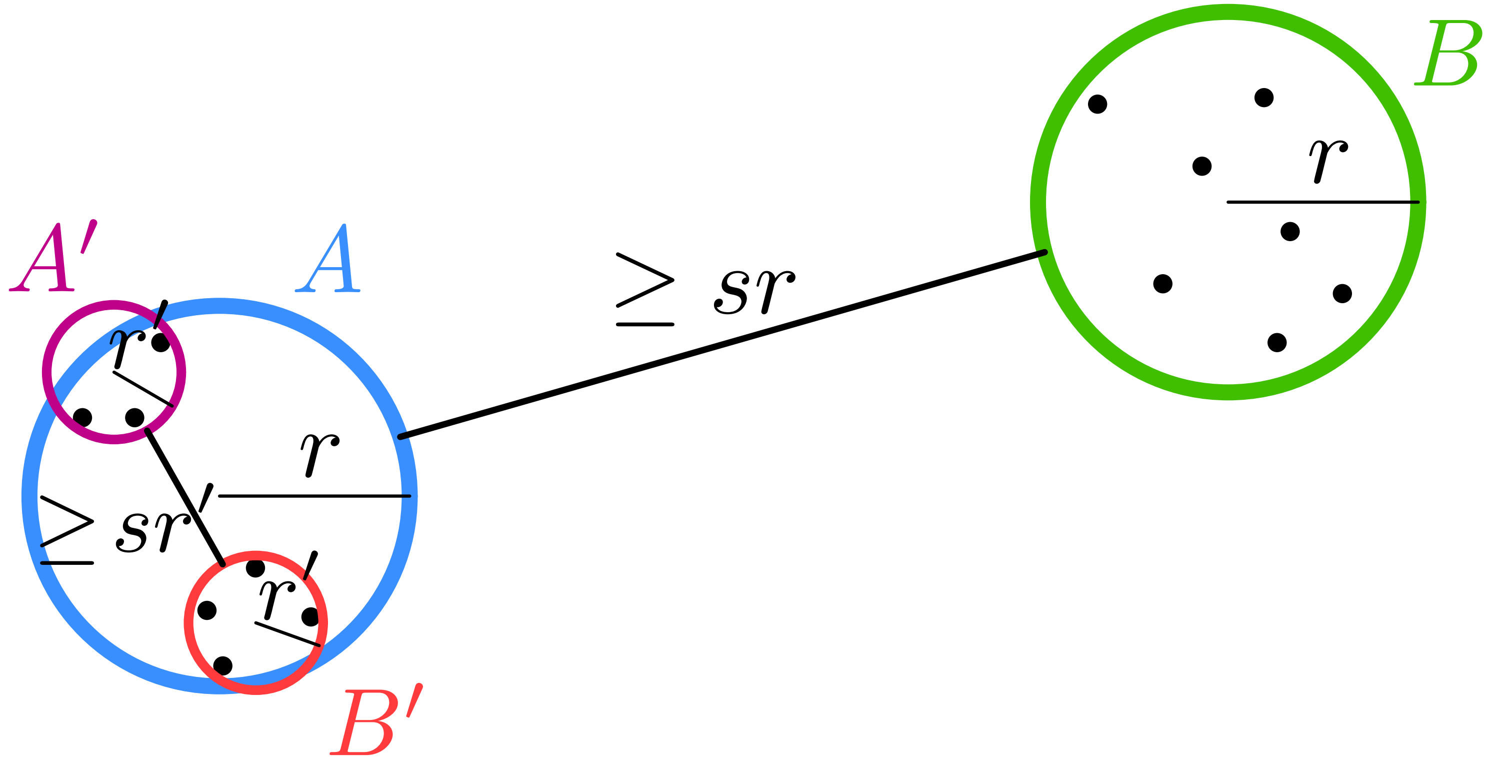
Well-Separated Pairs

Definition: A pair of disjoint point sets A and B in $\mathbb{R}^d$ is called s-well separated for an $s > 0$, if A and B both can be covered by a ball of radius r and the distance between the balls is at least sr.



Well-Separated Pairs

Definition: A pair of disjoint point sets A and B in $\mathbb{R}^d$ is called s-well separated for an $s > 0$, if A and B both can be covered by a ball of radius r and the distance between the balls is at least sr.



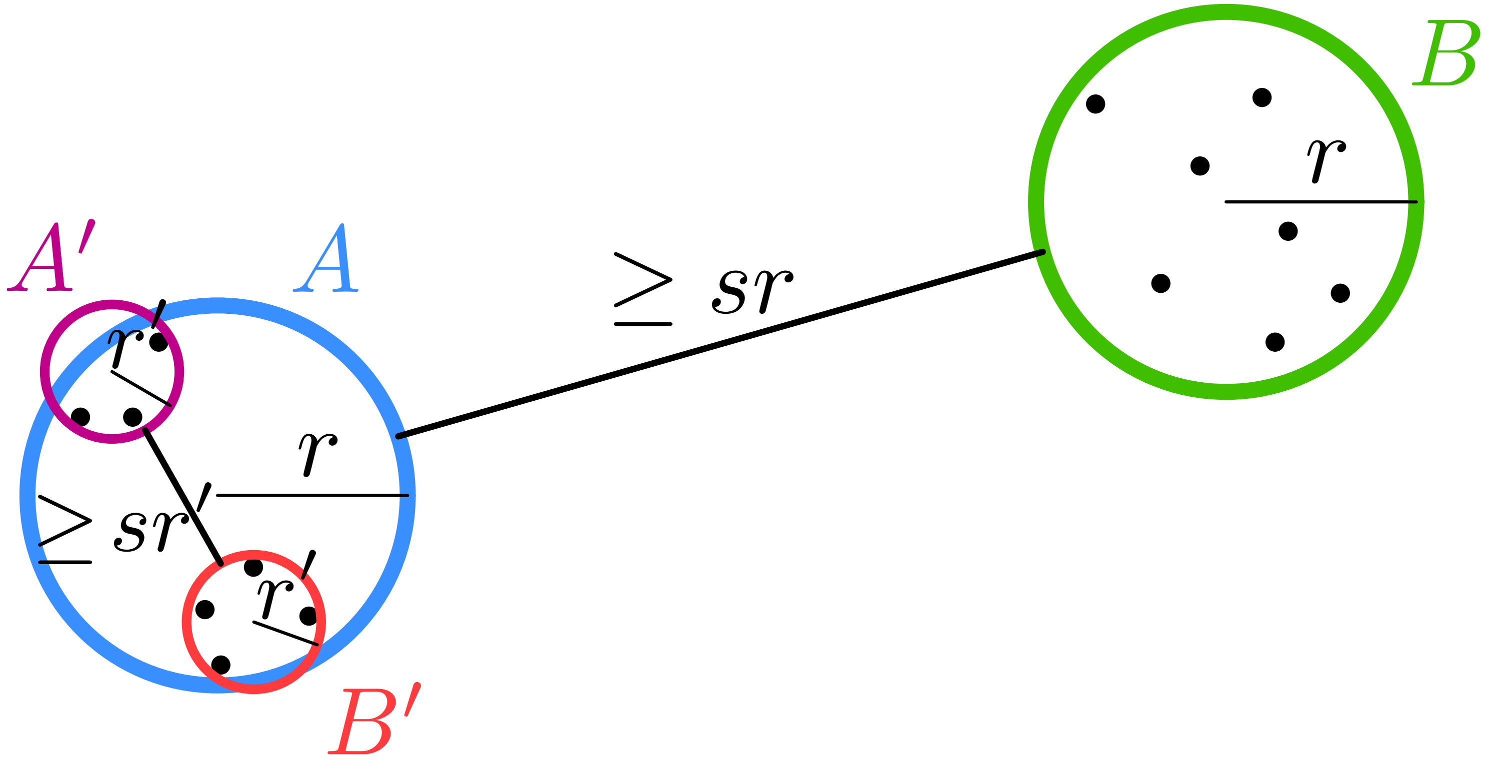
Observation:

- s-well separated $\Rightarrow$ s'-well separated for all $s' \leq s$
- singletons $\{a\}$ and $\{b\}$ are s-well separated for all $s > 0$

Well-Separated Pairs

Definition: A pair of disjoint point sets A and B in $\mathbb{R}^d$ is called s-well separated for an $s > 0$, if A and B both can be covered by a ball of radius r and the distance between the balls is at least sr.

Note: book uses $1/\varepsilon$ here.



Observation:

- s-well separated $\Rightarrow$ s'-well separated for all $s' \leq s$
- singletons $\{a\}$ and $\{b\}$ are s-well separated for all $s > 0$

Well-Separated Pair Decomposition

For a well-separated pair $\{A, B\}$ the distance between all point pairs in $A \otimes B := \{\{a, b\} \mid a \in A, b \in B, a \neq b\}$ is similar.

Goal: $o(n^2)$-data structure that approximates all $\binom{n}{2}$ pairwise distances of a point set $P = \{p_1, \ldots, p_n\}$.

Well-Separated Pair Decomposition

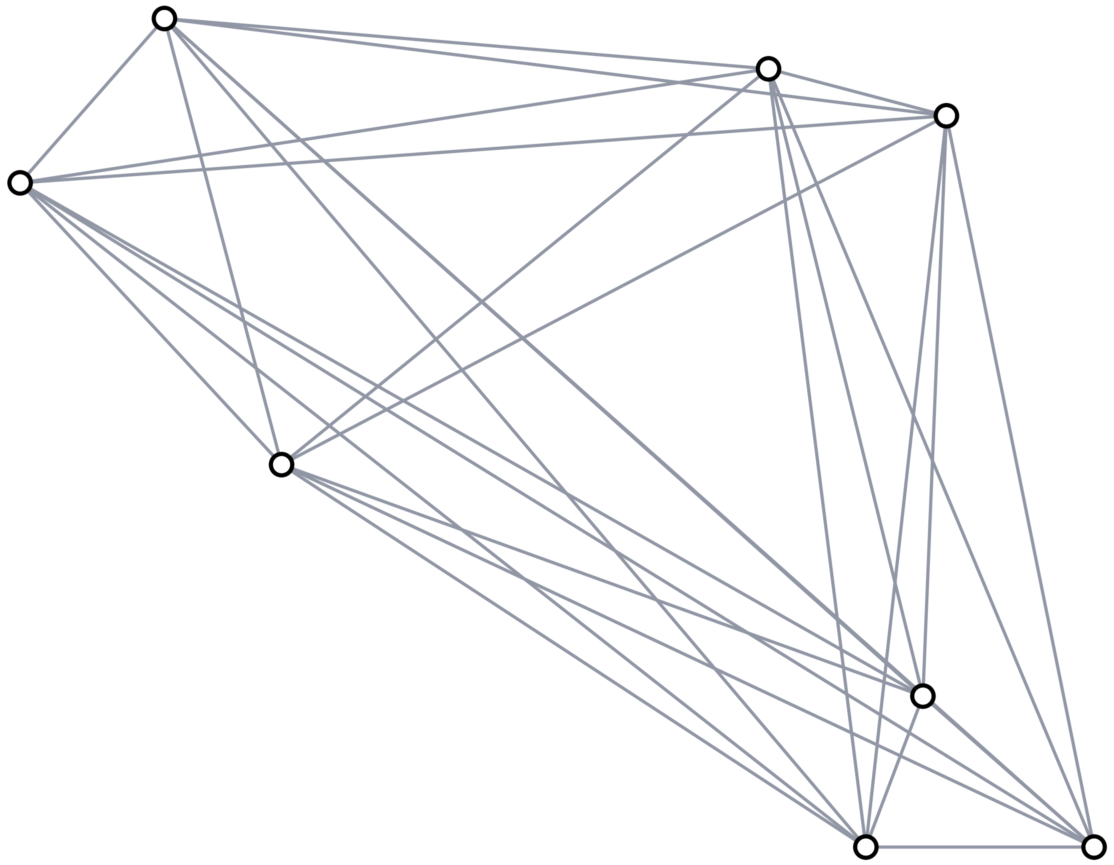
For a well-separated pair $\{A, B\}$ the distance between all point pairs in $A \otimes B := \{\{a, b\} \mid a \in A, b \in B, a \neq b\}$ is similar.

Goal: $o(n^2)$-data structure that approximates all $\binom{n}{2}$ pairwise distances of a point set $P = \{p_1, \ldots, p_n\}$.

Definition: For a set of points P and $s > 0$ an s-well separated pair decomposition (s-WSPD) is a set of pairs $\{\{A_1, B_1\}, \ldots, \{A_m, B_m\}\}$ with

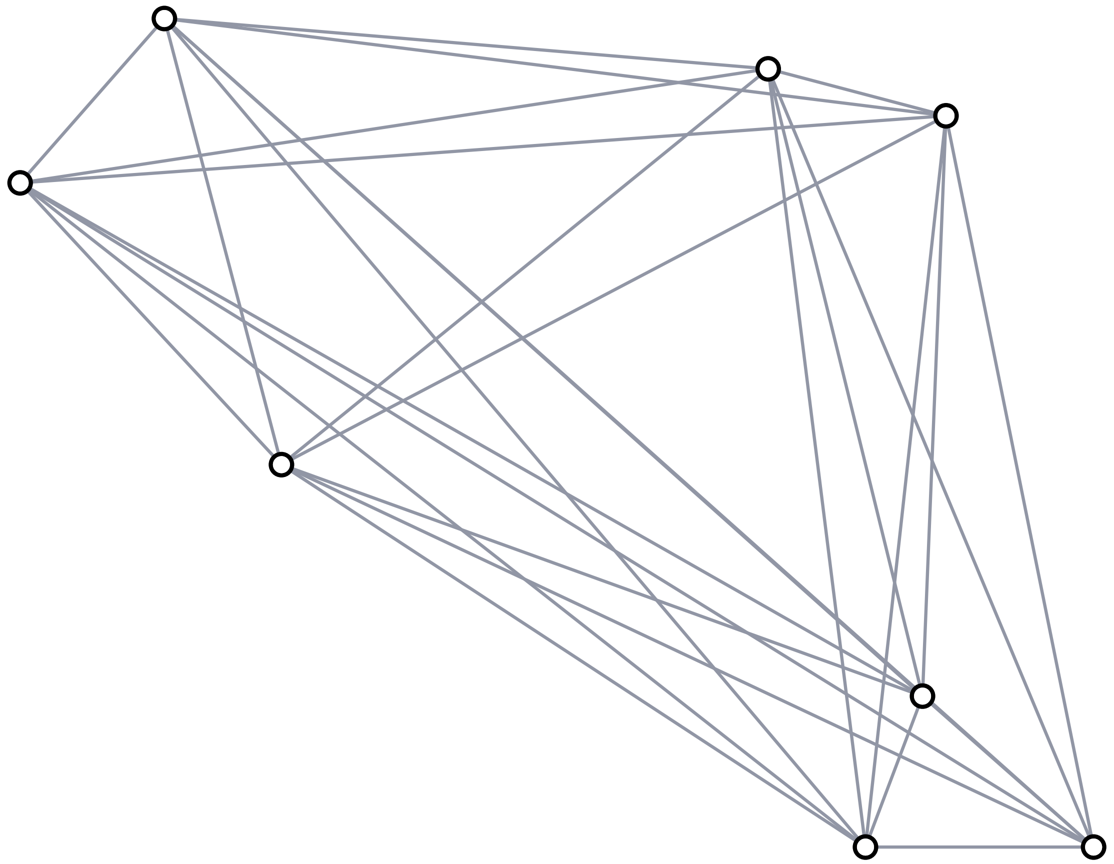
- $A_i, B_i \subset P$ for all i
- $A_i \cap B_i = \emptyset$ for all i
- $\bigcup_{i=1}^{m} A_i \otimes B_i = P \otimes P$
- $\{A_i, B_i\}$ s-well separated for all i

Example

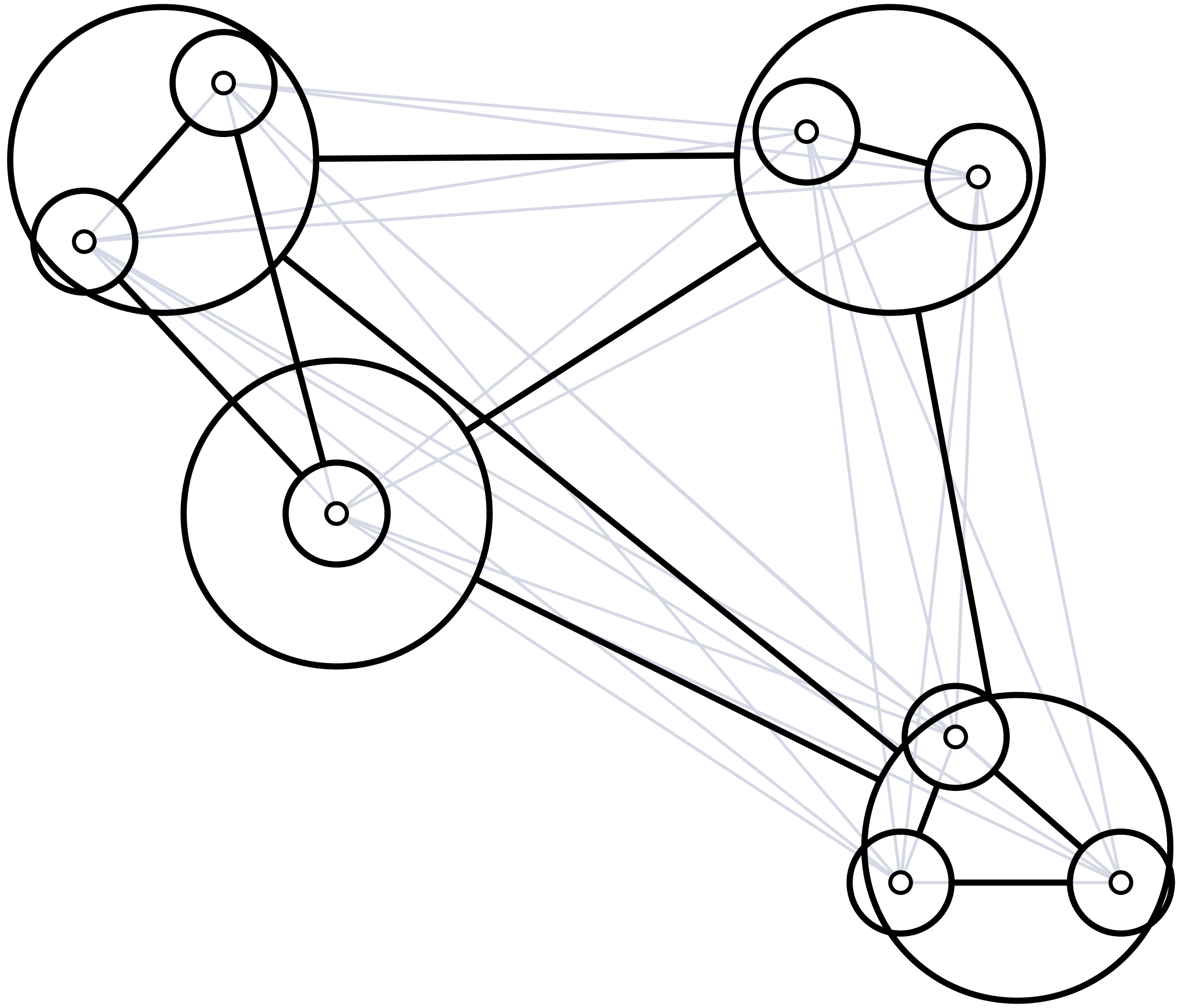


28 pairs of points

Example

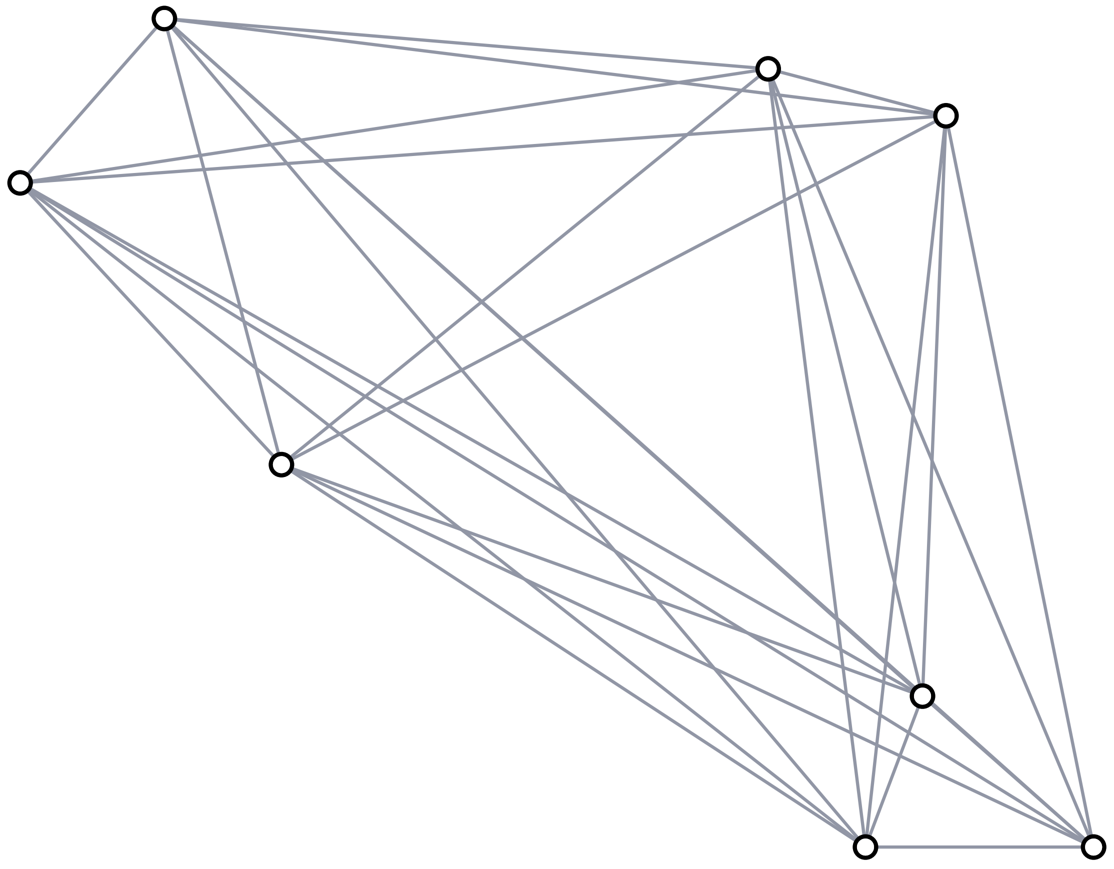


28 pairs of points

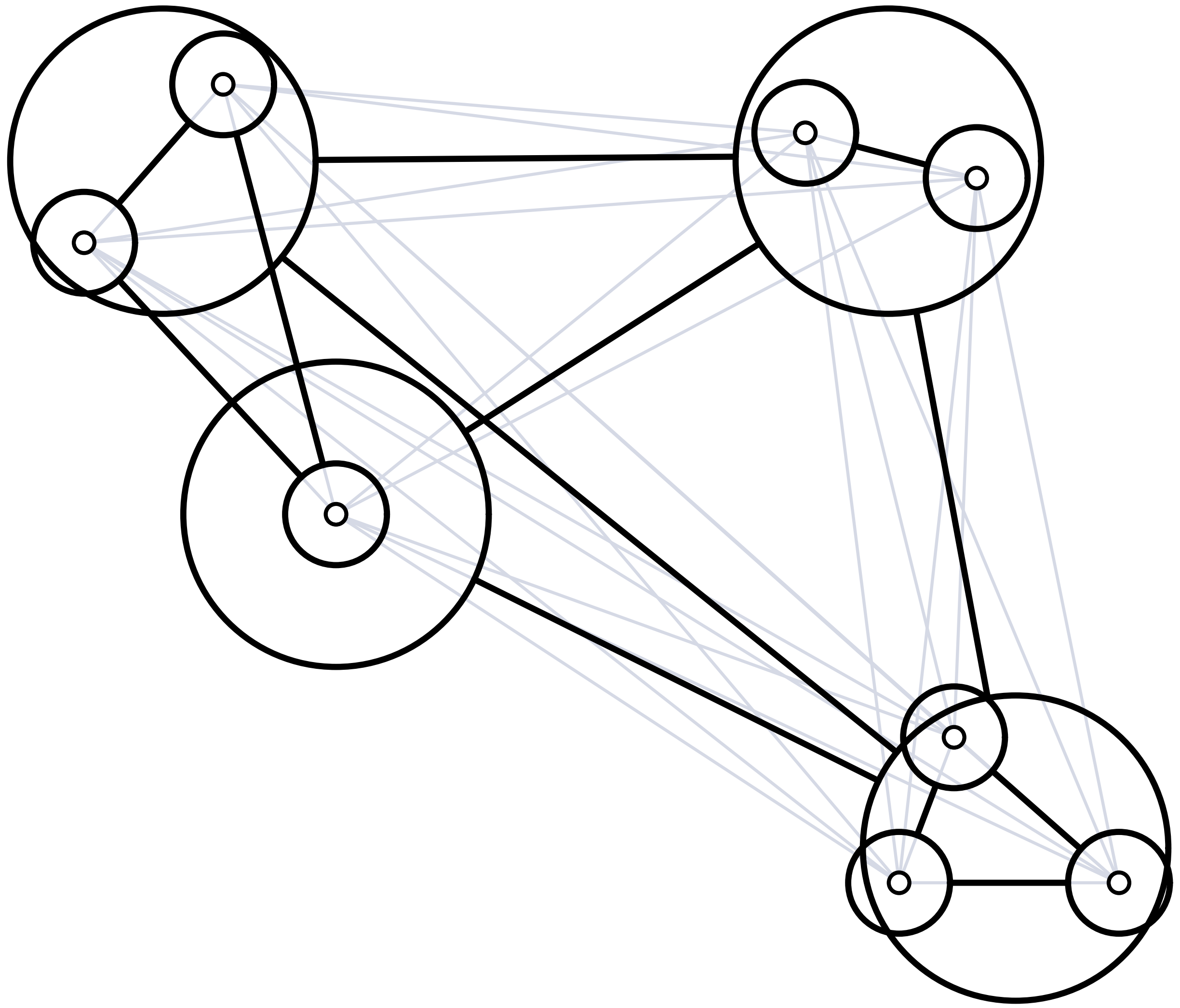


12 s-well separated pairs

Example



28 pairs of points



12 s-well separated pairs

WSPD of size $O(n^2)$ is trivial.

What is the 'size'? Can we get size $O(n)$?

Quiz

What size does a 2-WSPD on the following point set have at least?

A: 3

B: 4

C: 5

D: 6

Quiz

What size does a 2-WSPD on the following point set have at least?

A: 3

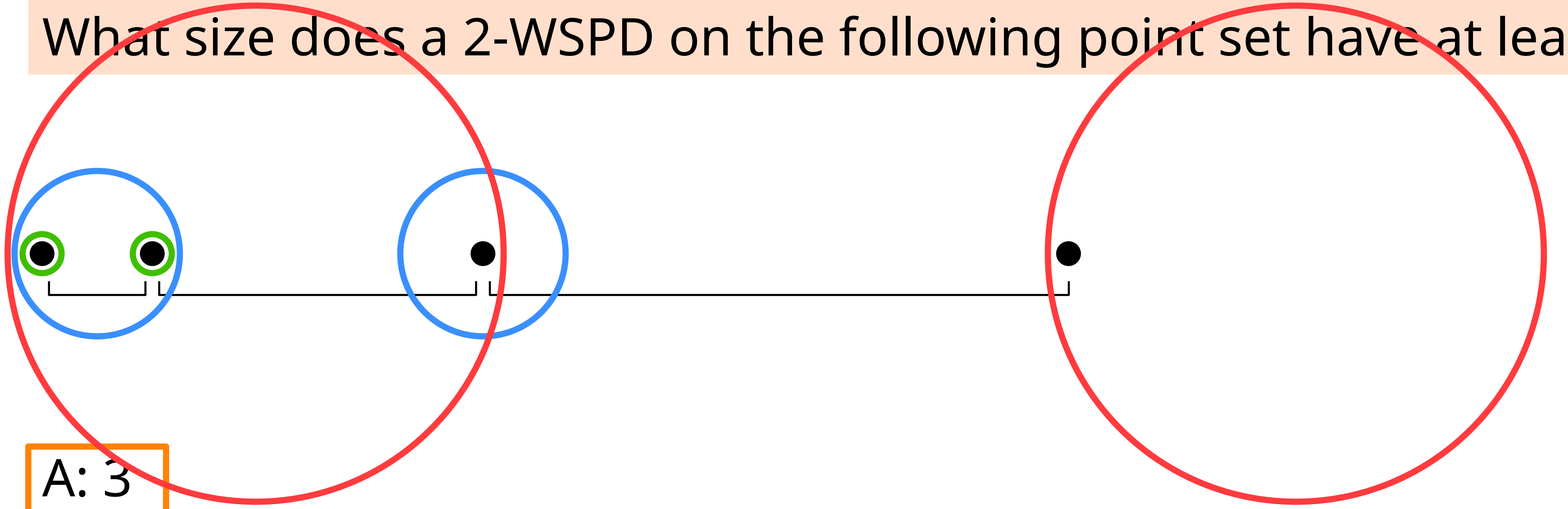
B: 4

C: 5

D: 6

Quiz

What size does a 2-WSPD on the following point set have at least?



A: 3

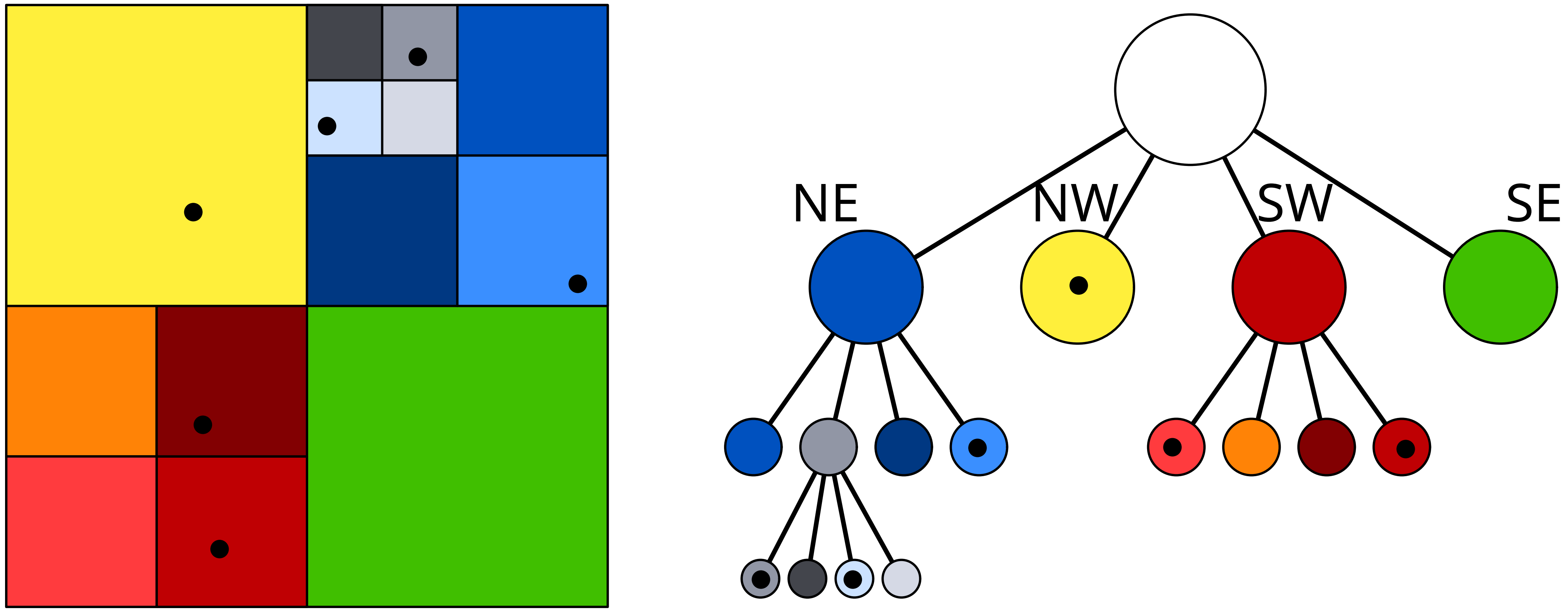
B: 4

C: 5

D: 6

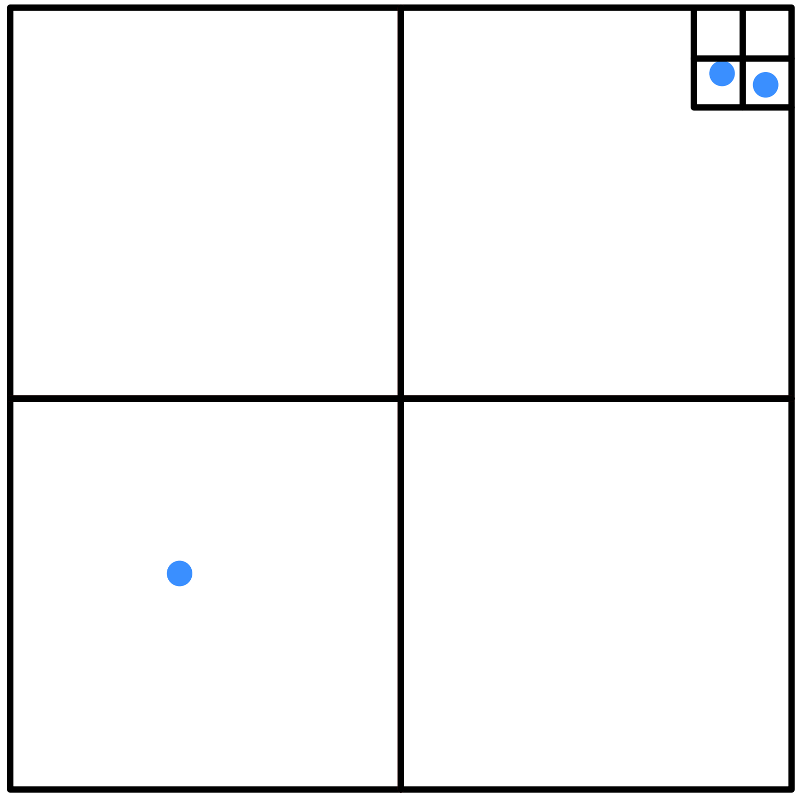
Reminder: quadtrees

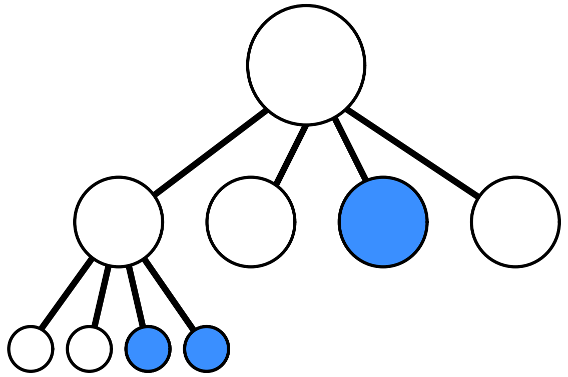
Definition: A quadtree is a rooted tree, in which every interior node has 4 children. Every node corresponds to a square, and the squares of children are the quadrants of the parent's square.



Reminder: Compressed quadtrees

Definition: A compressed quadtree is a quadtree in which paths of non-separating inner nodes are compressed to an edge.

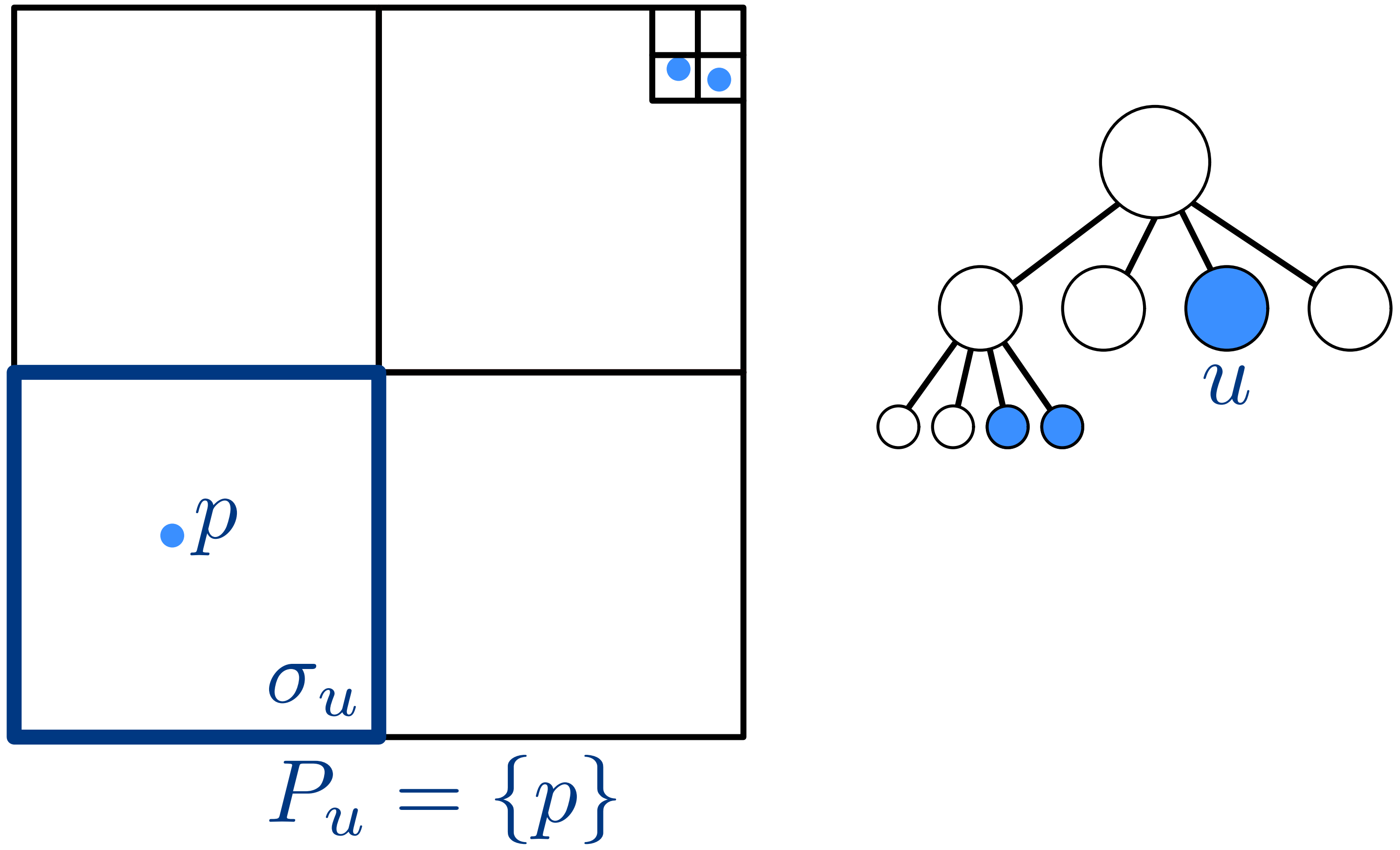




Theorem 2: A compressed quadtree for n points in $\mathbb{R}^d$ for fixed d has size $O(n)$ and can be computed in $O(n \log n)$ time.

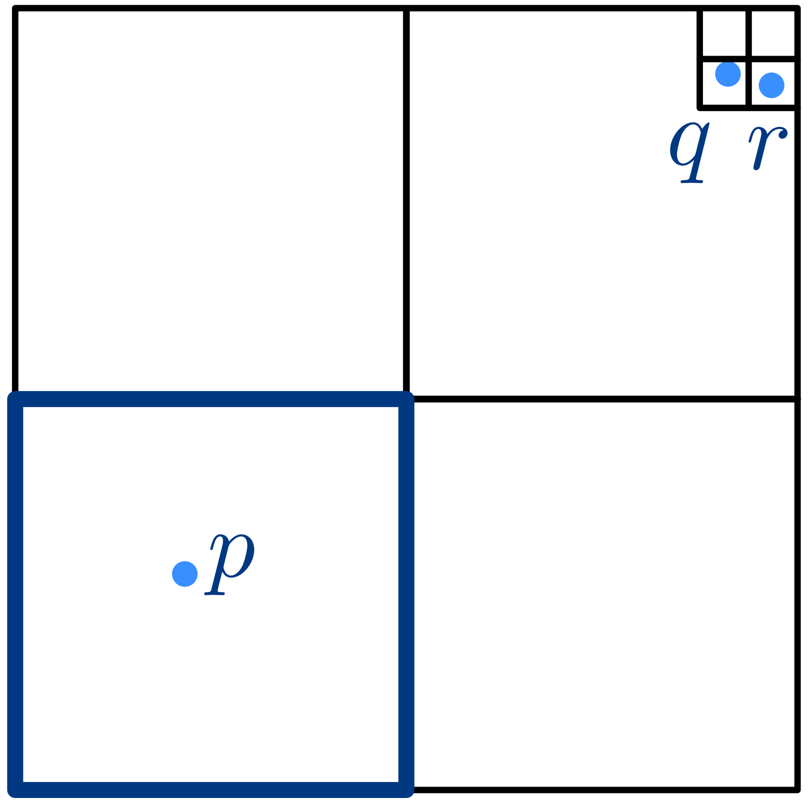
Representative and Level

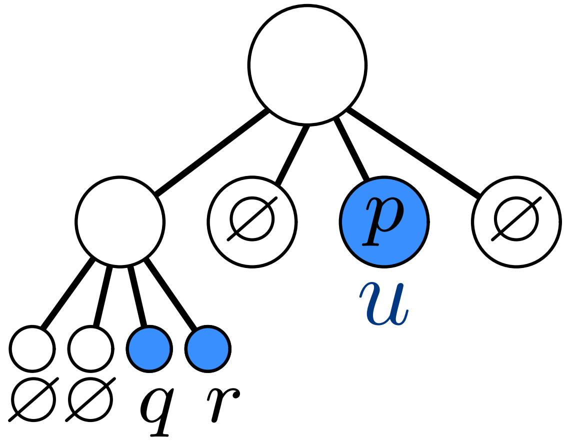
Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.



Representative and Level

Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.



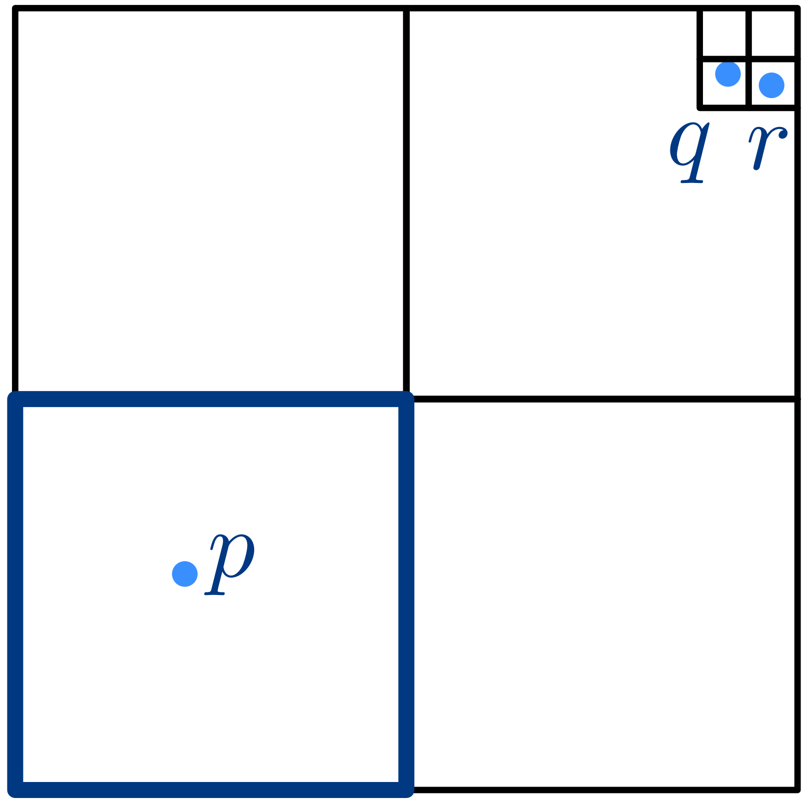


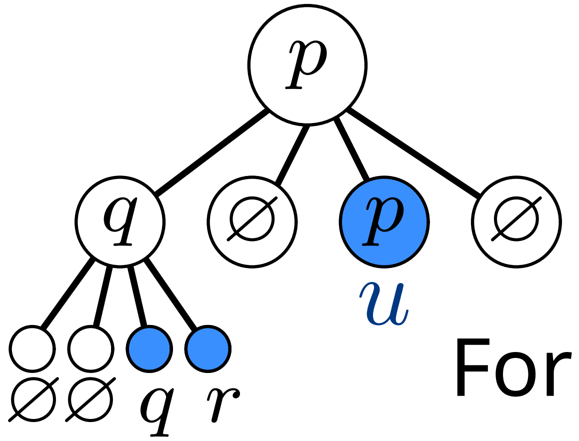
For every leaf u define the representative

$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \text{ (}u\text{ is leaf)} \\ \emptyset & \text{otherwise.} \end{cases}$$

Representative and Level

Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.





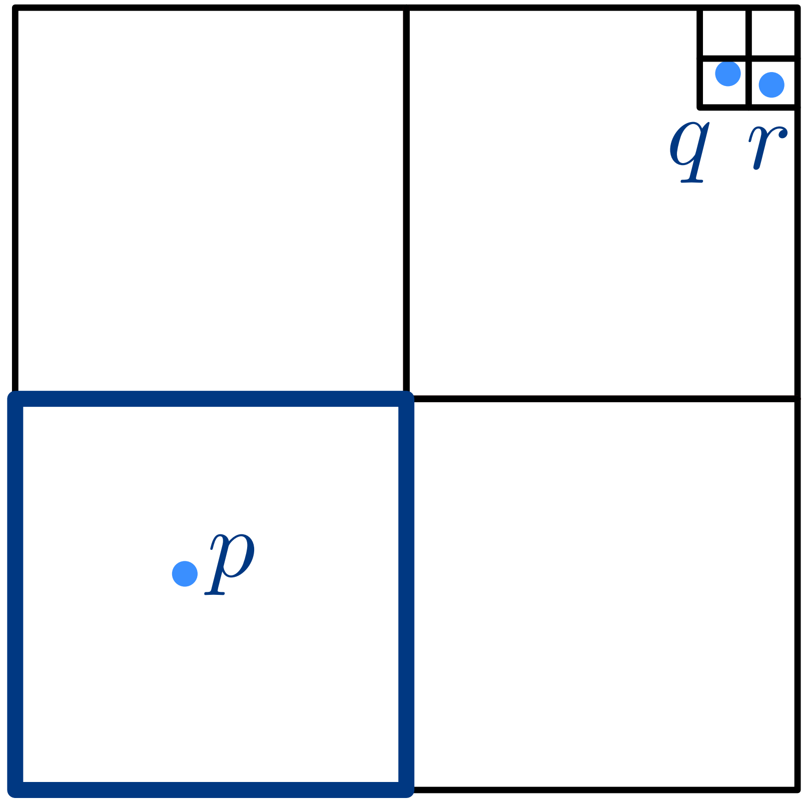
For every leaf u define the representative

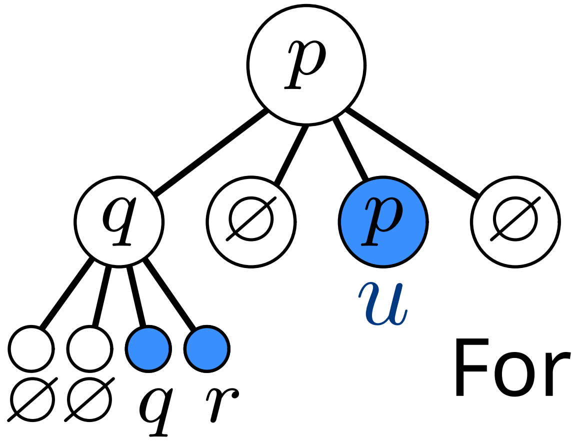
$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \ (u \text{ is leaf}) \\ \varnothing & \text{otherwise.} \end{cases}$$

For every inner node v set
$\text{rep}(v) = \text{rep}(u)$ of a non-empty child u of v.

Representative and Level

Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.





For every leaf u define the representative

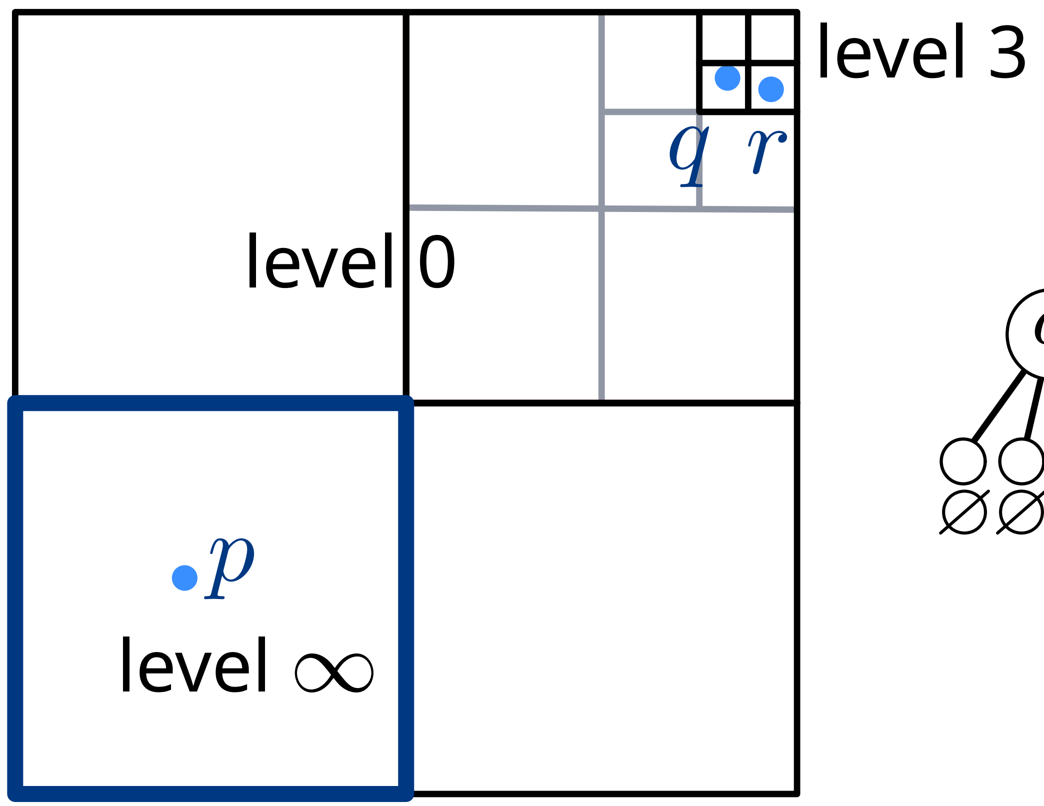
$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \text{ (}u\text{ is leaf)} \\ \varnothing & \text{otherwise.} \end{cases}$$

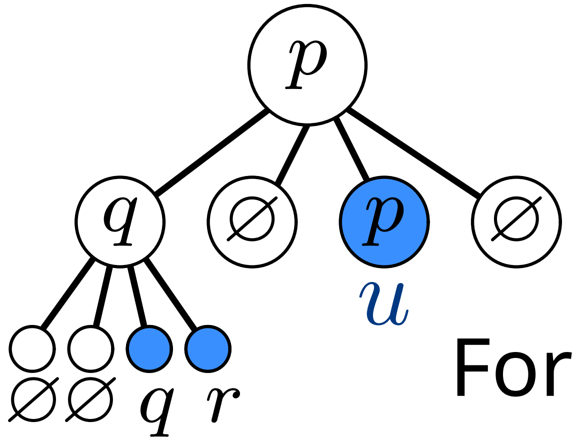
For every inner node v set
$\text{rep}(v) = \text{rep}(u)$ of a non-empty child u of v.

Definition: For every inner node u of a quadtree $\mathcal{T}(P)$ let level(u) be the level of u in the corresponding uncompressed quadtree. For leaves u, level$(u) := \infty$

Representative and Level

Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.





For every leaf u define the representative

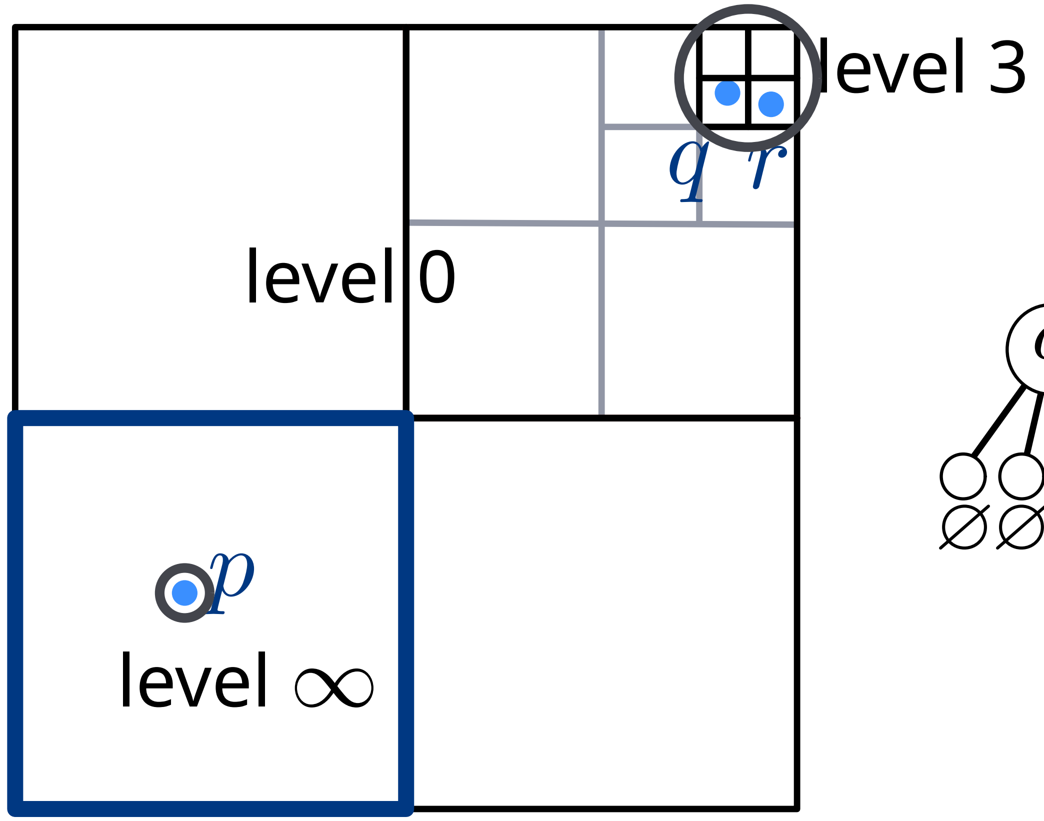
$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \text{ (}u\text{ is leaf)} \\ \varnothing & \text{otherwise.} \end{cases}$$

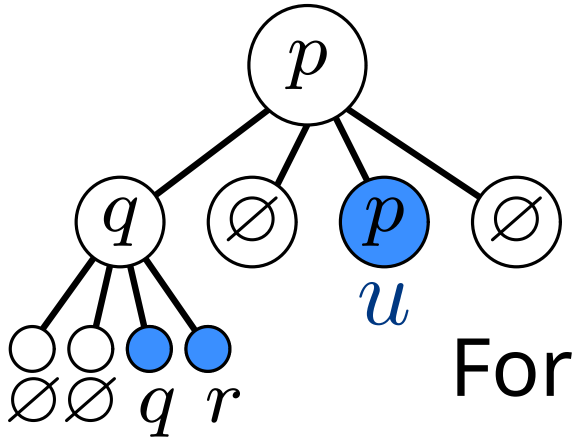
For every inner node v set $\text{rep}(v) = \text{rep}(u)$ of a non-empty child u of v.

Definition: For every inner node u of a quadtree $\mathcal{T}(P)$ let level(u) be the level of u in the corresponding uncompressed quadtree. For leaves u, level$(u) := \infty$

Representative and Level

Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.





For every leaf u define the representative

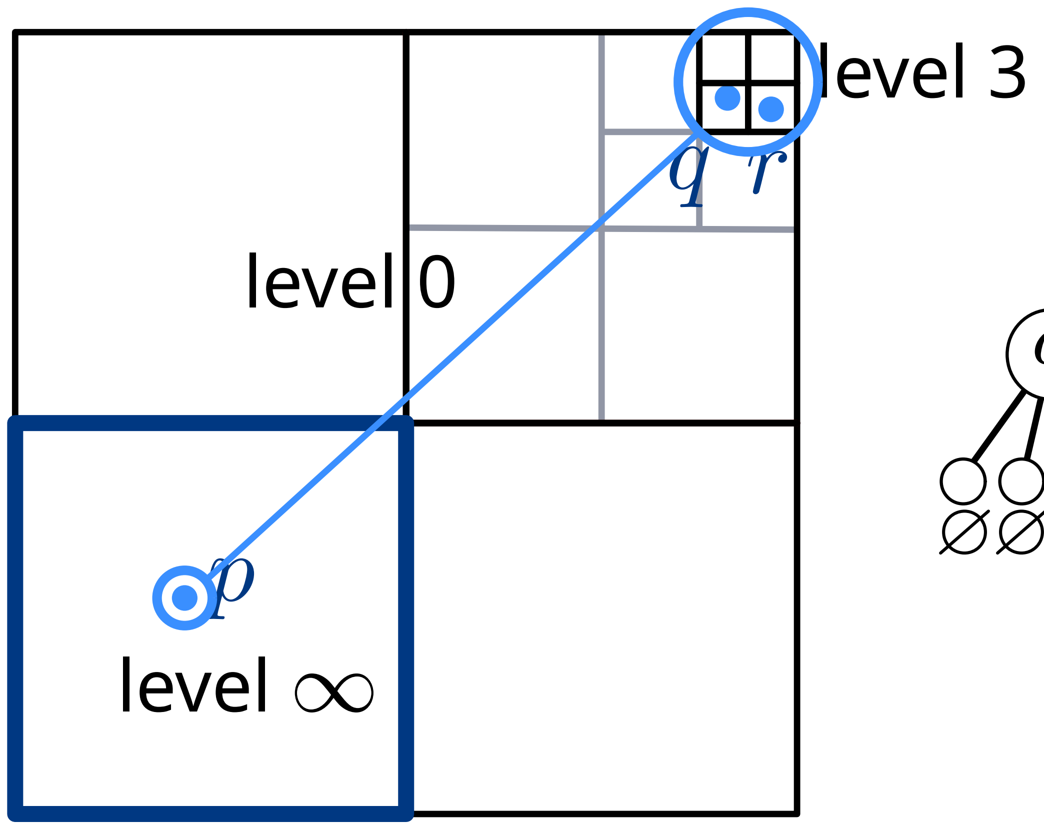
$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \text{ (}u\text{ is leaf)} \\ \varnothing & \text{otherwise.} \end{cases}$$

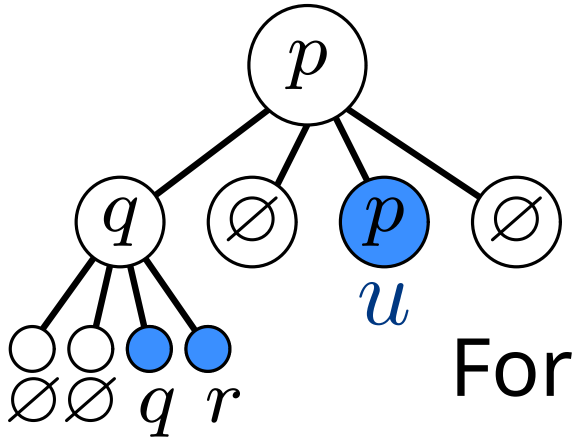
For every inner node v set
$\text{rep}(v) = \text{rep}(u)$ of a non-empty child u of v.

Notes: (a) levels in book < 0, (b) book works with $\Delta(u)$ = radius of circle around square (or 0 for leaves) instead.

Representative and Level

Definition: For every node u of a quadtree $\mathcal{T}(P)$ let $P_u = \sigma_u \cap P$, where σ_u is the square corresponding to u.





For every leaf u define the representative

$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \ (u \text{ is leaf}) \\ \varnothing & \text{otherwise.} \end{cases}$$

For every inner node v set
$\text{rep}(v) = \text{rep}(u)$ of a non-empty child u of v.

next: using quadtree to compute WSPD

Well-Separated Pair Decomposition

Construction

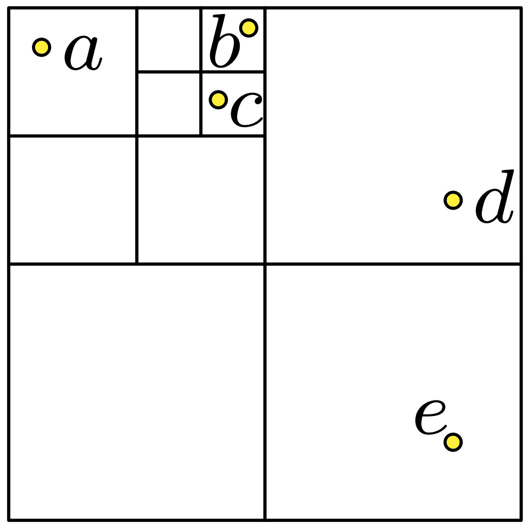
Construction of a WSPD

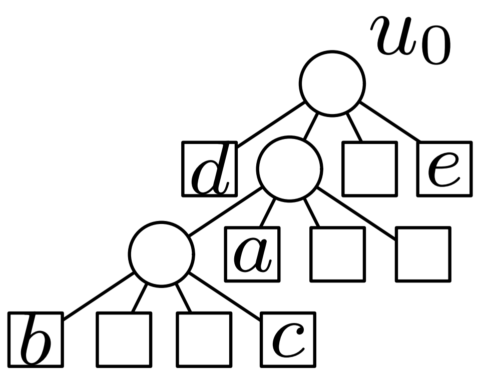
wsPairs$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

1: **if** rep$(u) = \emptyset$ or rep$(v) = \emptyset$ or leaves $u = v$ **then return** $\emptyset$
2: **else if** P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
3: **else**
4: **if** level$(u) >$ level(v) **then** exchange u and v
5: $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
6: **return** $\bigcup_{i=1}^{m}$ wsPairs$(u_i, v, \mathcal{T}, s)$





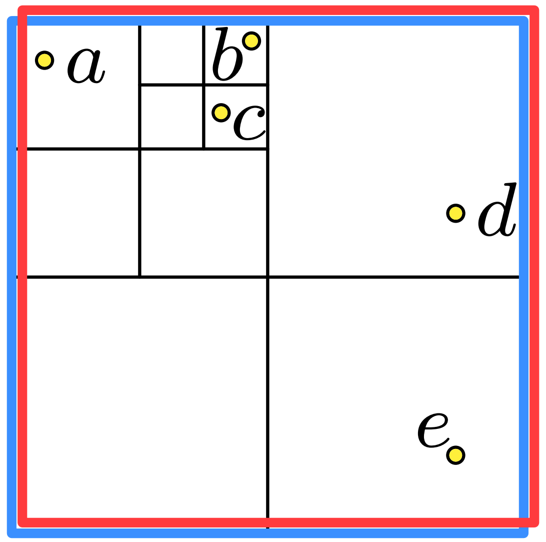
Construction of a WSPD

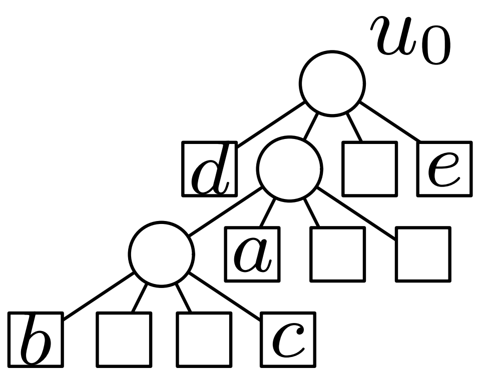
$\textsc{wsPairs}(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
    if level$(u) >$ level(v) **then** exchange u and v
    $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
    return $\bigcup_{i=1}^{m}$ $\textsc{wsPairs}(u_i, v, \mathcal{T}, s)$





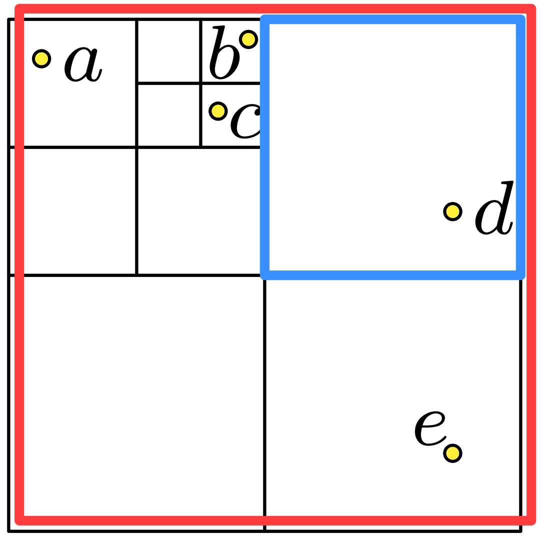
Construction of a WSPD

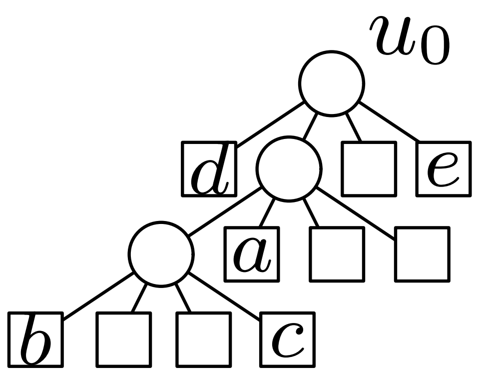
wsPairs$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

1: **if** rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
2: **else if** P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
3: **else**
4: **if** level$(u) >$ level(v) **then** exchange u and v
5: $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
6: **return** $\bigcup_{i=1}^{m}$ wsPairs$(u_i, v, \mathcal{T}, s)$





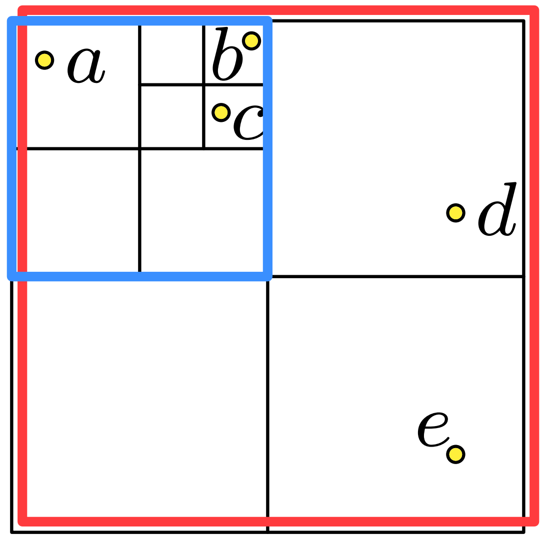
Construction of a WSPD

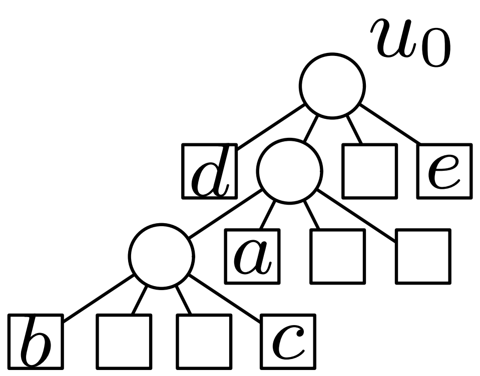
WSPAIRS$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

```
if rep(u) = ∅ or rep(v) = ∅ or leaves u = v then return ∅
else if P_u and P_v s-well separated then return {{u, v}}
else
    if level(u) > level(v) then exchange u and v
    (u_1, ..., u_m) ← children of u in T
    return ⋃_{i=1}^{m} WSPAIRS(u_i, v, T, s)
```





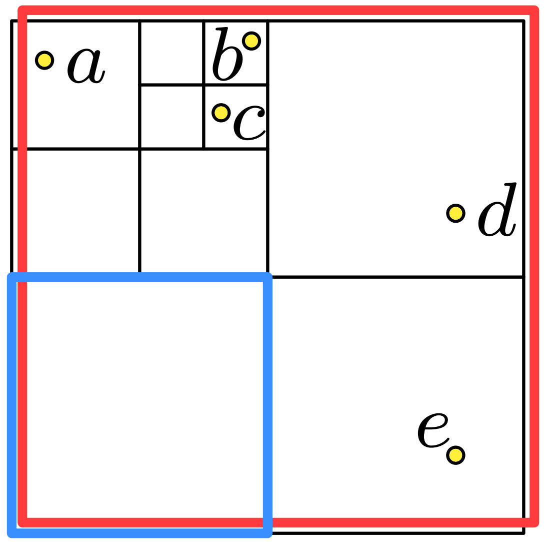
Construction of a WSPD

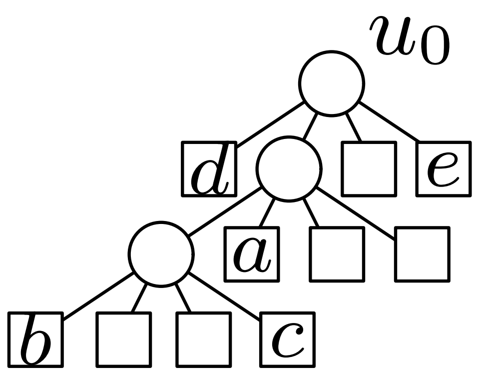
$\textsc{wsPairs}(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
    if level$(u) >$ level(v) **then** exchange u and v
    $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
    return $\bigcup_{i=1}^{m} \textsc{wsPairs}(u_i, v, \mathcal{T}, s)$





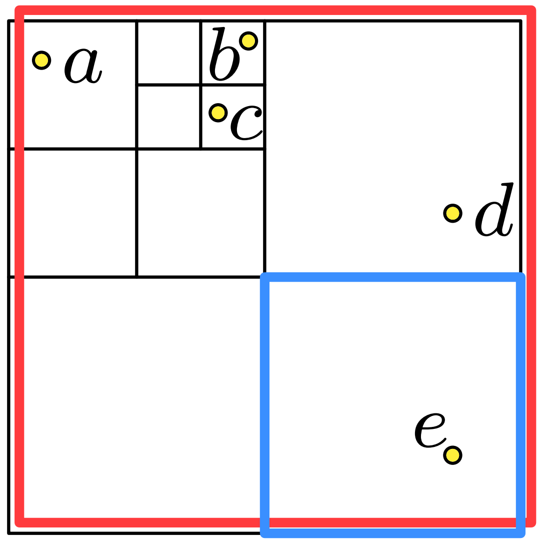
Construction of a WSPD

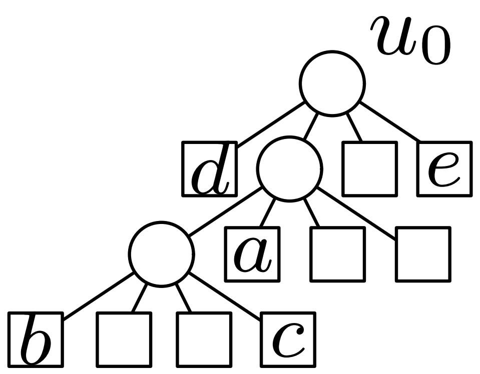
wsPairs$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
    if level$(u) >$ level(v) **then** exchange u and v
    $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
    return $\bigcup_{i=1}^{m}$ wsPairs$(u_i, v, \mathcal{T}, s)$



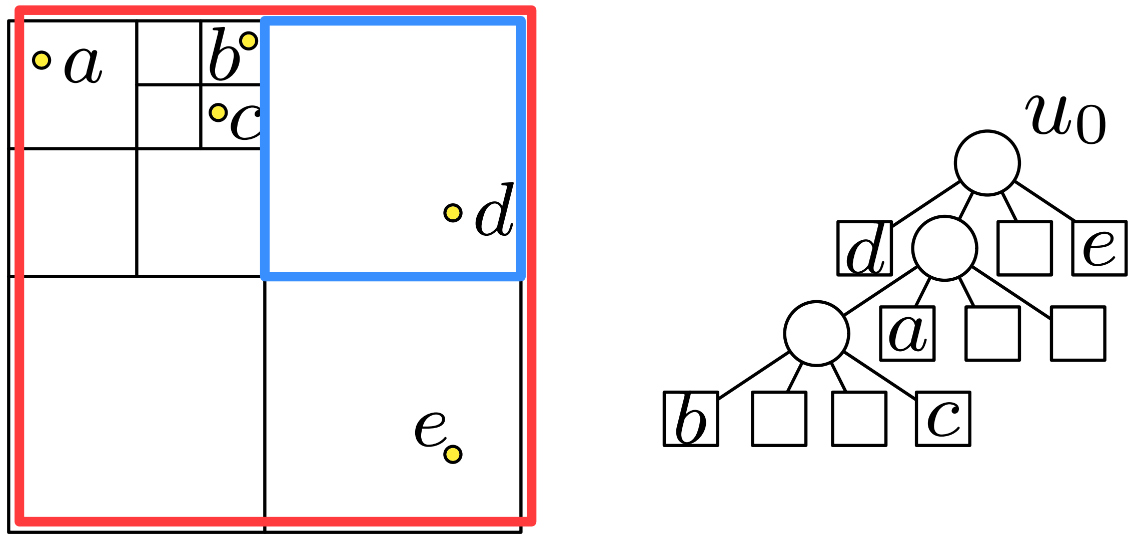


Construction of a WSPD

$\textsc{wsPairs}(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$
Output: WSPD for $P_u \otimes P_v$

1: **if** rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
2: **else if** P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
3: **else**
4:     **if** level$(u) >$ level(v) **then** exchange u and v
5:     $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
6:     **return** $\bigcup_{i=1}^{m}$ $\textsc{wsPairs}(u_i, v, \mathcal{T}, s)$



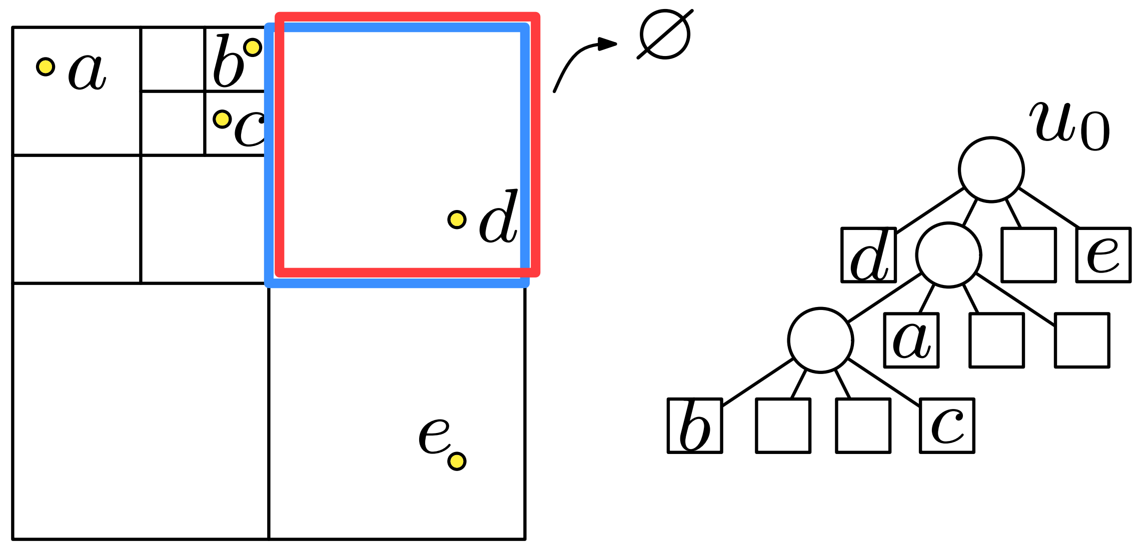
Construction of a WSPD

WSPAIRS$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

```
if rep(u) = ∅ or rep(v) = ∅ or leaves u = v then return ∅
else if P_u and P_v s-well separated then return {{u, v}}
else
    if level(u) > level(v) then exchange u and v
    (u_1, ..., u_m) ← children of u in T
    return ⋃_{i=1}^{m} WSPAIRS(u_i, v, T, s)
```



Construction of a WSPD

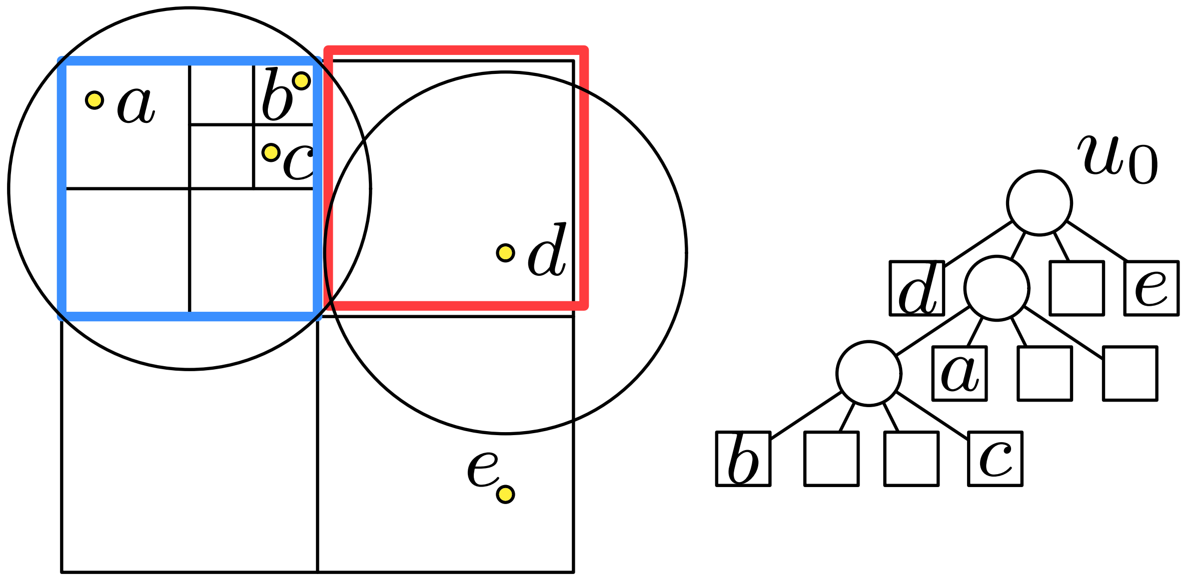
$\textsc{wsPairs}(u, v, \mathcal{T}, s)$

Input: quadtree r

Output: WSPD fo

circles around σ_u and σ_v (or radius 0 for point in a leaf),
increase radius of smaller circle,
check distance $\geq sr$ in $O(1)$ time

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
  if level$(u) >$ level(v) **then** exchange u and v
  $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
  return $\bigcup_{i=1}^{m}$ $\textsc{wsPairs}(u_i, v, \mathcal{T}, s)$



Construction of a WSPD

wsPairs$(u, v, \mathcal{T}, s)$

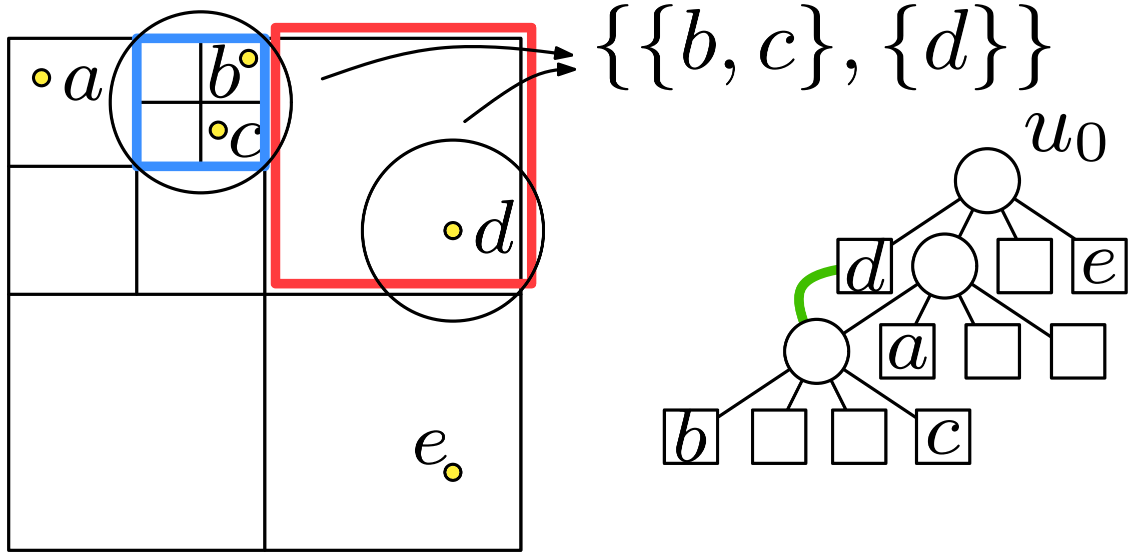
Input: quadtree r

Output: WSPD fo

circles around σ_u and σ_v (or radius 0 for point in a leaf),
increase radius of smaller circle,
check distance $\geq sr$ in $O(1)$ time

if rep$(u) = \emptyset$ or rep$(v) = \emptyset$ or leaves $u = v$ **then return** $\emptyset$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
  if level$(u) >$ level(v) **then** exchange u and v
  $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
  return $\bigcup_{i=1}^{m}$ wsPairs$(u_i, v, \mathcal{T}, s)$

$\{\{b, c\}, \{d\}\}$



Construction of a WSPD

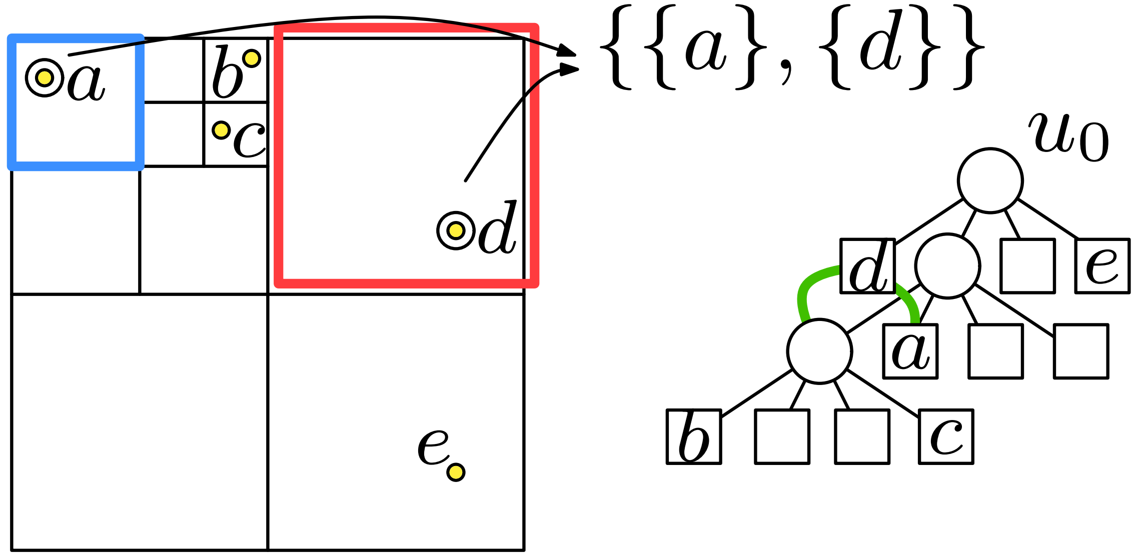
wsPairs$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
  if level$(u) >$ level(v) **then** exchange u and v
  $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
  return $\bigcup_{i=1}^{m}$ wsPairs$(u_i, v, \mathcal{T}, s)$

$\{\{b, c\}, \{d\}\}$
$\{\{a\}, \{d\}\}$



Construction of a WSPD

WSPAIRS$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$

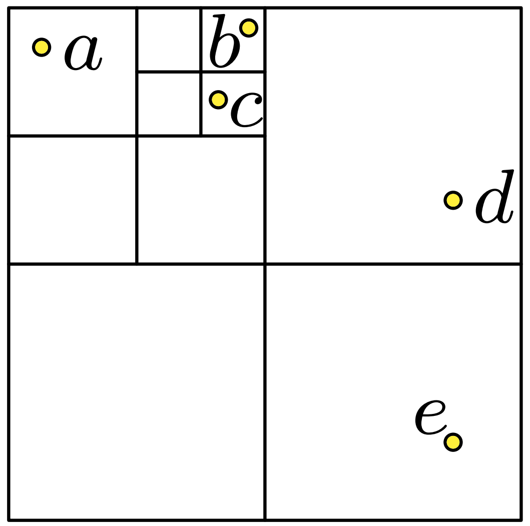
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$

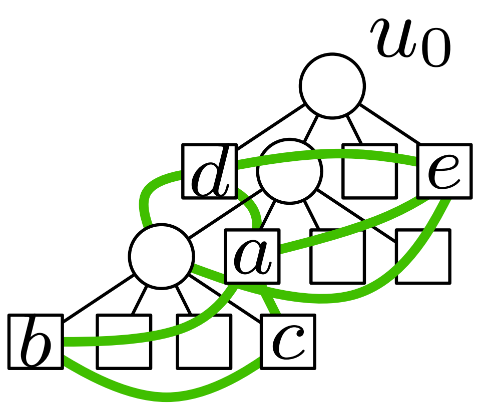
else

  if level$(u) >$ level(v) **then** exchange u and v

  $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$

  return $\bigcup_{i=1}^{m}$ WSPAIRS$(u_i, v, \mathcal{T}, s)$





$\{\{b, c\}, \{d\}\}$
$\{\{a\}, \{d\}\}$
$\{\{b, c\}, \{e\}\}$
$\{\{d\}, \{e\}\}$
$\{\{a\}, \{b\}\}$
$\{\{a\}, \{c\}\}$
$\{\{b\}, \{c\}\}$
$\{\{a\}, \{e\}\}$

Construction of a WSPD

wsPairs$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

Output: WSPD for $P_u \otimes P_v$

1: **if** rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
2: **else if** P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
3: **else**
4:     **if** level$(u) >$ level(v) **then** exchange u and v
5:     $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
6:     **return** $\bigcup_{i=1}^{m}$ wsPairs$(u_i, v, \mathcal{T}, s)$

- initial call wsPairs$(u_0, u_0, \mathcal{T}, s)$
- avoid duplicate wsPairs$(u_i, u_j, \mathcal{T}, s)$ and wsPairs$(u_j, u_i, \mathcal{T}, s)$
- pairs of leaves are s-well separated $\rightarrow$ algorithm terminates
- output are pairs of quadtree nodes

Quiz

Is the size of the s-WSPD constructed minimal?

A: Yes, because the s-WSPD is unique.

B: Yes, because all s-WSPDs have the same size.

C: No, not necessarily.

Quiz

Is the size of the s-WSPD constructed minimal?

A: Yes, because the s-WSPD is unique.

B: Yes, because all s-WSPDs have the same size.

$\boxed{\text{C: No, not necessarily.}}$

Quiz

Is the size of the s-WSPD constructed minimal?

A: Yes, because the s-WSPD is unique.

B: Yes, because all s-WSPDs have the same size.

C: No, not necessarily.

Question: How many pairs are generated by the algorithm?

Well-Separated Pair Decomposition

Complexity

Analysis of WSPD-Construction

Theorem: For a point set P in $\mathbb{R}^d$ and $s \geq 1$ we can construct an s-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

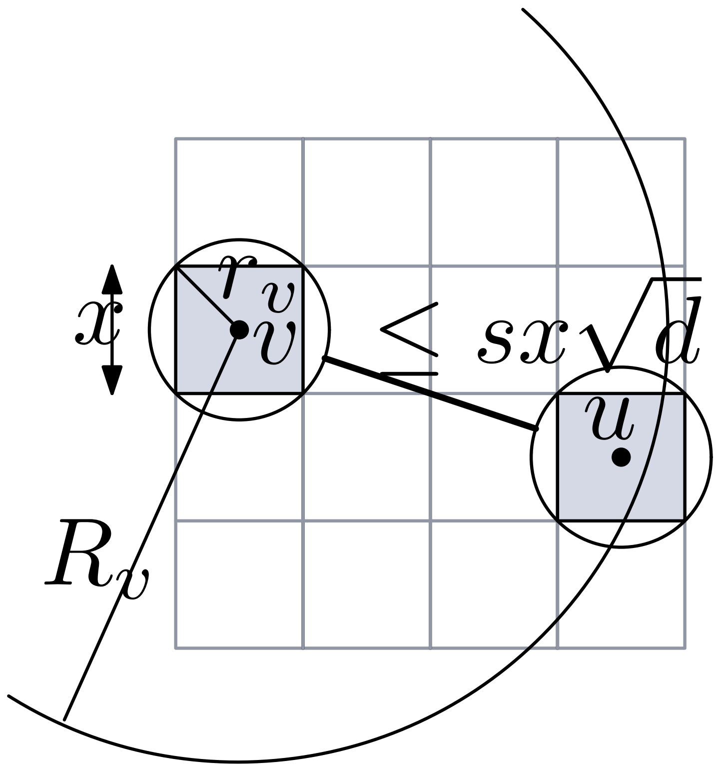
Analysis of WSPD-Construction

Theorem: For a point set P in $\mathbb{R}^d$ and $s \geq 1$ we can construct an s-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

Proof sketch:

Assumptions: $s \geq 1$, QT uncompressed.

Count the non-terminal calls.



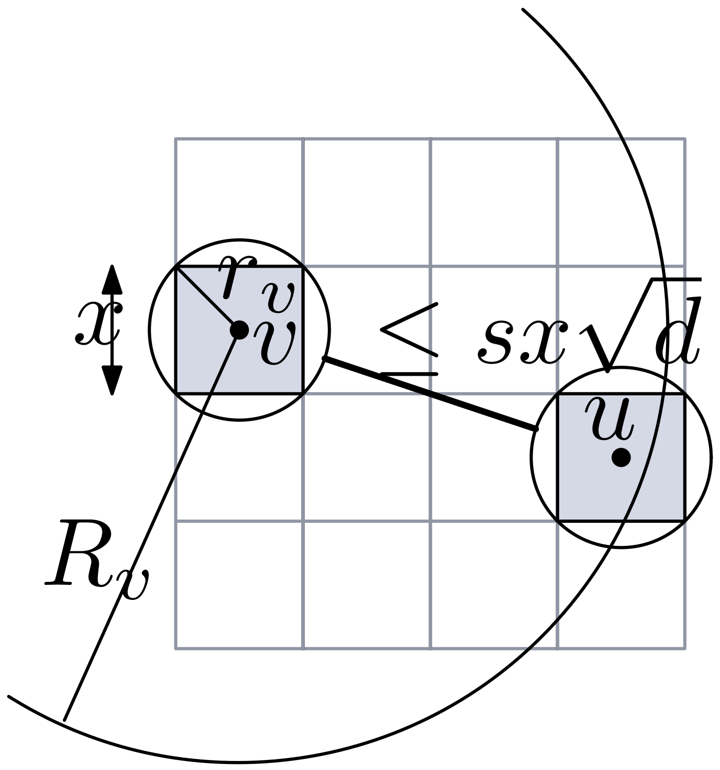
Analysis of WSPD-Construction

Theorem: For a point set P in $\mathbb{R}^d$ and $s \geq 1$ we can construct an s-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

Proof sketch:

Assumptions: $s \geq 1$, QT uncompressed.

Count the non-terminal calls.



WSPAIRS$(u, v, \mathcal{T}, s)$

Input: quadtree nodes u, v, quadtree $\mathcal{T}$, $s > 0$

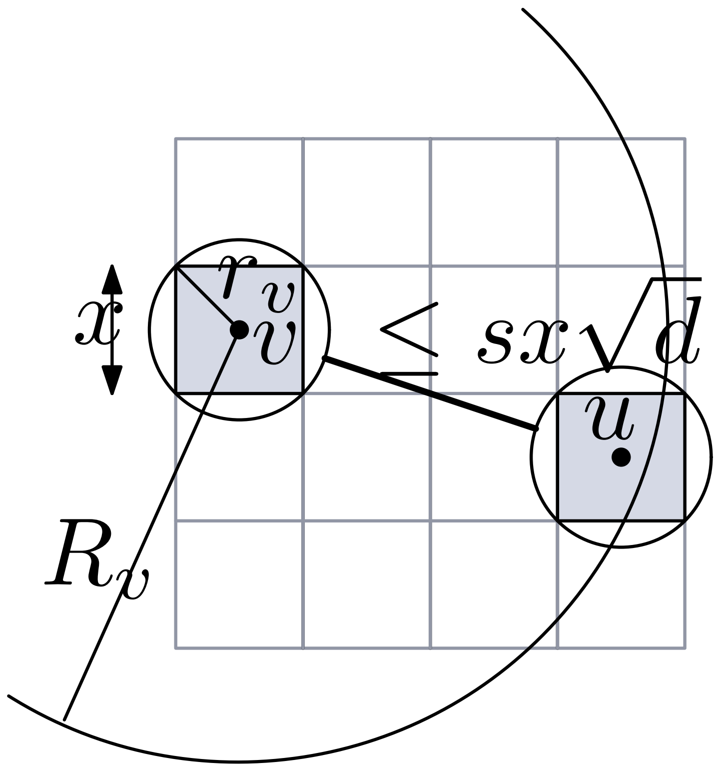
Output: WSPD for $P_u \otimes P_v$

if rep$(u) = \varnothing$ or rep$(v) = \varnothing$ or leaves $u = v$ **then return** $\varnothing$
else if P_u and P_v s-well separated **then return** $\{\{u, v\}\}$
else
    if level$(u) >$ level(v) **then** exchange u and v
    $(u_1, \ldots, u_m) \leftarrow$ children of u in $\mathcal{T}$
    return $\bigcup_{i=1}^{m}$ WSPAIRS$(u_i, v, \mathcal{T}, s)$

Analysis of WSPD-Construction

Theorem: For a point set P in $\mathbb{R}^d$ and $s \geq 1$ we can construct an s-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

Proof sketch:



Assumptions: $s \geq 1$, QT uncompressed.

Count the non-terminal calls.

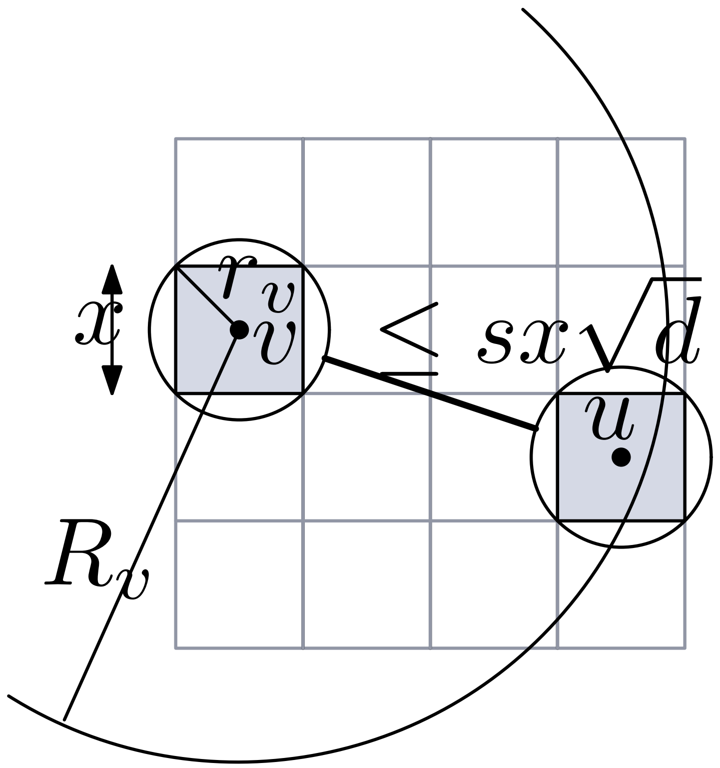
Charging argument: charge non-term. call to the non-split square.

claim: $O(s^d)$ charges to each square

Analysis of WSPD-Construction

Theorem: For a point set P in $\mathbb{R}^d$ and $s \geq 1$ we can construct an s-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

Proof sketch:



Assumptions: $s \geq 1$, QT uncompressed.

Count the non-terminal calls.

Charging argument: charge non-term. call to the non-split square.

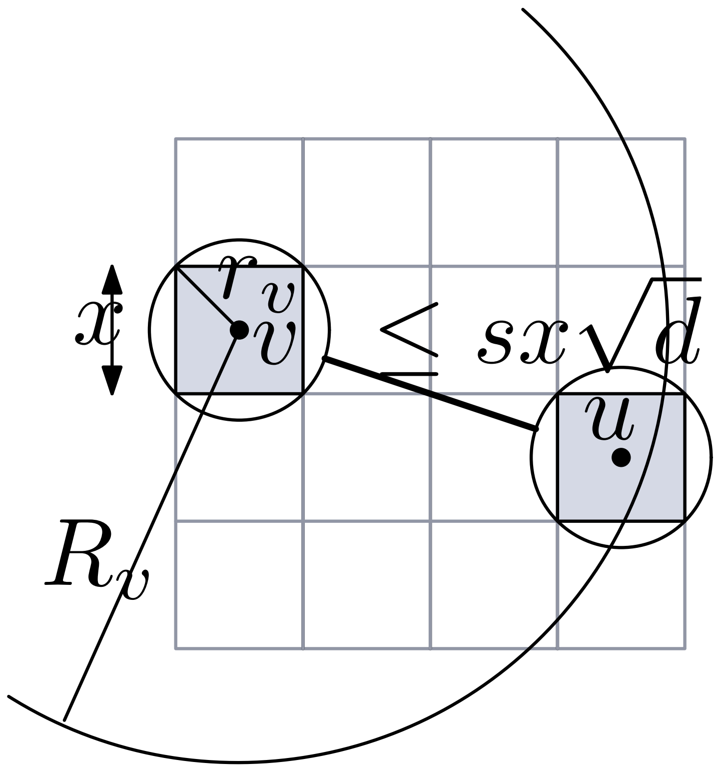
claim: $O(s^d)$ charges to each square

Consider call (u, v) with v smaller of side length x.

Analysis of WSPD-Construction

Theorem: For a point set P in $\mathbb{R}^d$ and $s \geq 1$ we can construct an s-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

Proof sketch:



Assumptions: $s \geq 1$, QT uncompressed.

Count the non-terminal calls.

Charging argument: charge non-term. call to the non-split square.

claim: $O(s^d)$ charges to each square

Consider call (u, v) with v smaller of side length x.

u, v are not separated,

u is at most factor 2 larger than v

$\Rightarrow$ distance between the balls

$\leq s \max(r_u, r_v) \leq 2sr_v = sx\sqrt{d}$

$\Rightarrow$ distance between their centers

$\leq (1/2 + 1 + s)x\sqrt{d} \leq 3sx\sqrt{d} =: R_v$

packing lemma: only $O(s^d)$ such squares.

Packing Lemma

Lemma: Let B be a ball of radius r in $\mathbb{R}^d$ and X a set of pairwise disjoint quadtree cells with side length $\geq x$, that intersect B. Then

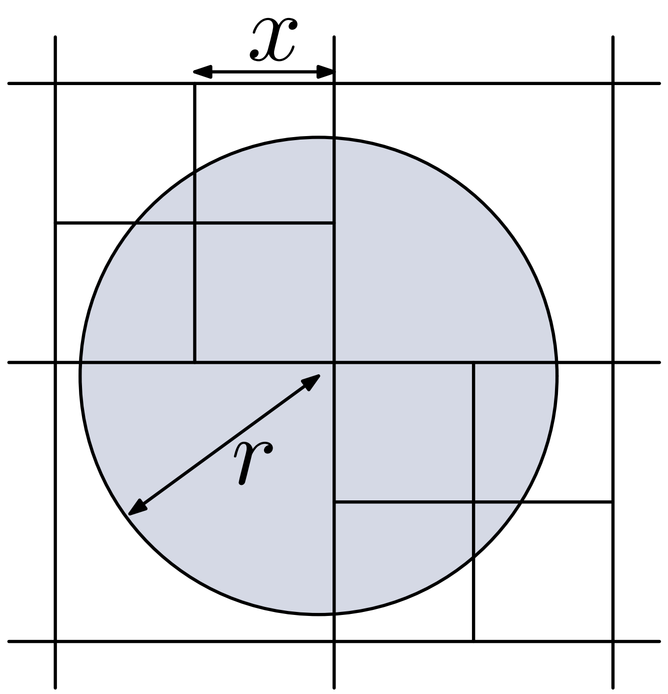
$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$

Packing Lemma

Lemma: Let B be a ball of radius r in $\mathbb{R}^d$ and X a set of pairwise disjoint quadtree cells with side length $\geq x$, that intersect B. Then

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$

Proof:

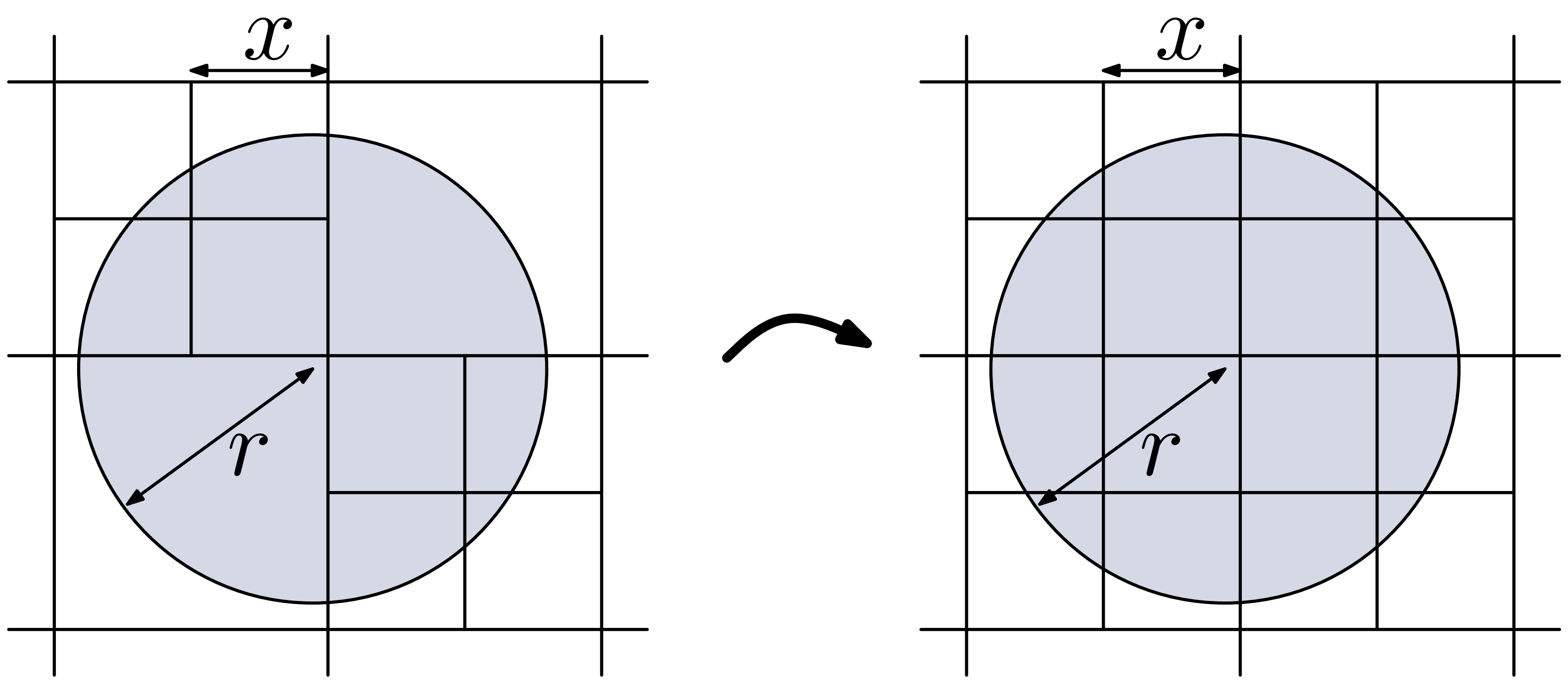


Packing Lemma

Lemma: Let B be a ball of radius r in $\mathbb{R}^d$ and X a set of pairwise disjoint quadtree cells with side length $\geq x$, that intersect B. Then

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$

Proof:

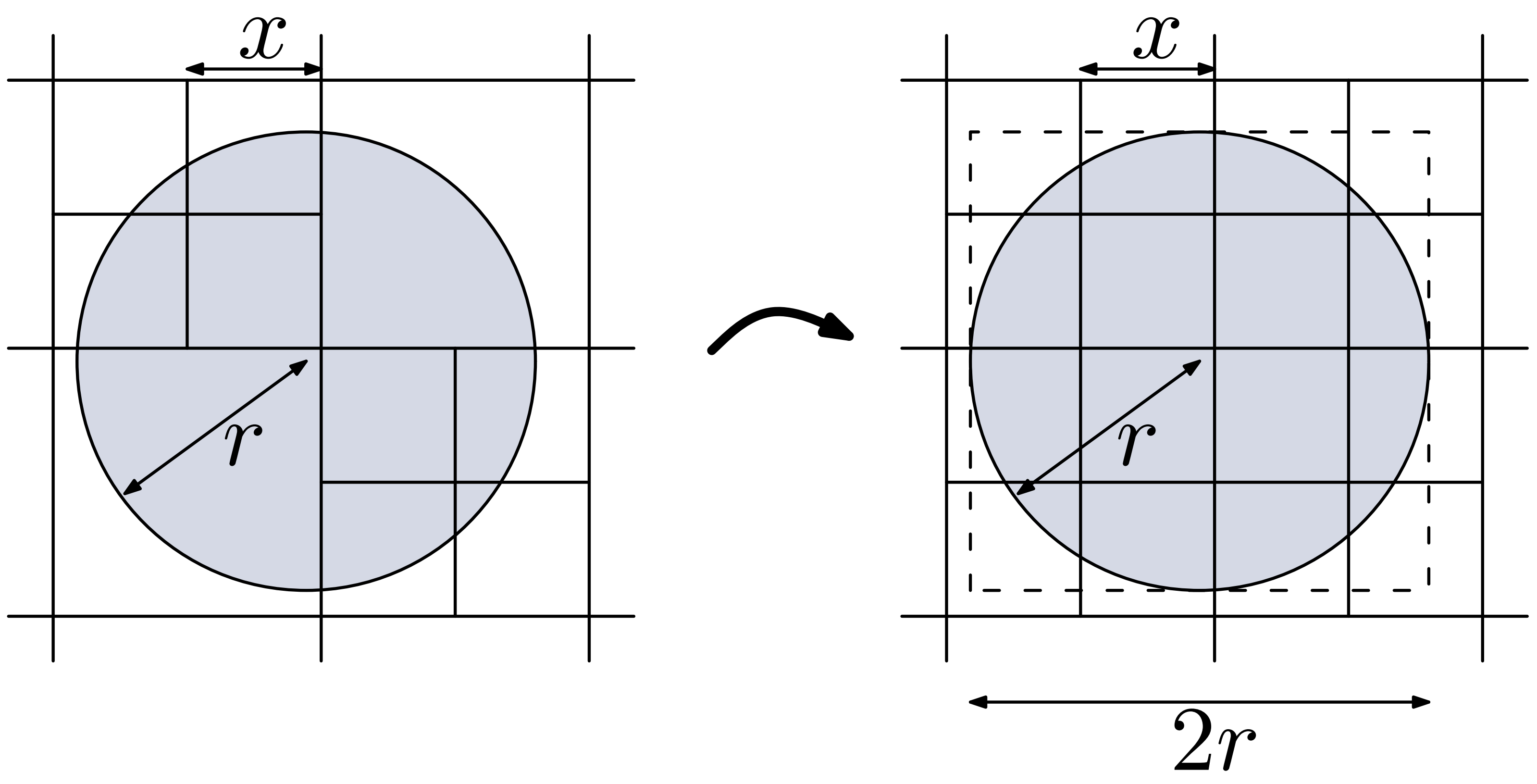


Packing Lemma

Lemma: Let B be a ball of radius r in $\mathbb{R}^d$ and X a set of pairwise disjoint quadtree cells with side length $\geq x$, that intersect B. Then

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$

Proof:



in every dimension at most $1 + \lceil 2r/x \rceil$ squares can intersect the ball

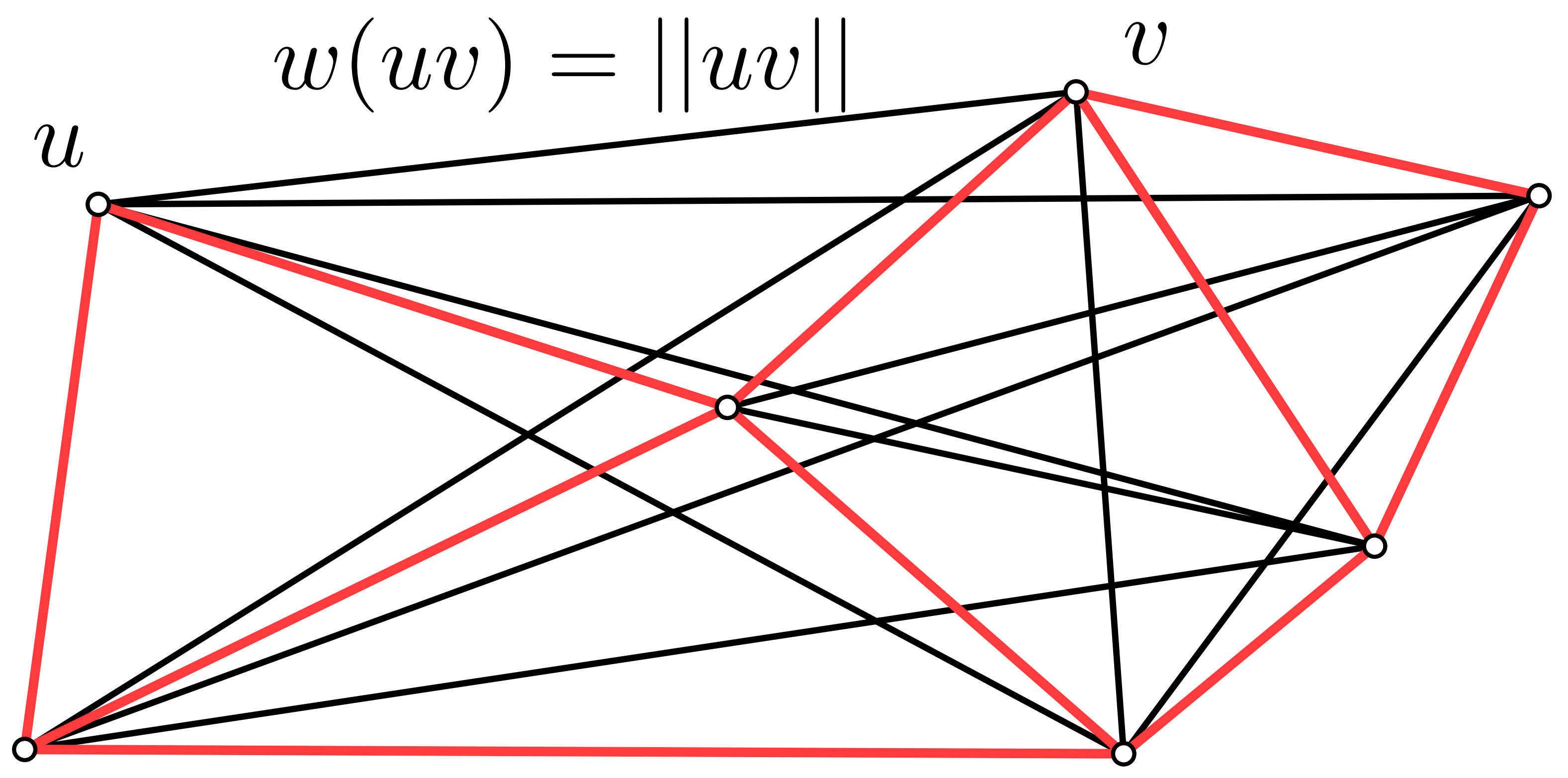
Well-Separated Pair Decomposition

Application: t-spanner

t-spanner

For a set P of n points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{EG}(P) = (P, \binom{P}{2})$ is the complete, weighted graph with Euclidean distances as edge weights.

Since $\mathcal{EG}(P)$ has $\Theta(n^2)$ edges, we want a sparse graph with $O(n)$ edges such that the shortest paths in the graph approximate the edge weights of $\mathcal{EG}(P)$.



t-spanner

For a set P of n points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{EG}(P) = (P, \binom{P}{2})$ is the complete, weighted graph with Euclidean distances as edge weights.

Since $\mathcal{EG}(P)$ has $\Theta(n^2)$ edges, we want a sparse graph with $O(n)$ edges such that the shortest paths in the graph approximate the edge weights of $\mathcal{EG}(P)$.

Definition: A weighted graph G with vertex set P is called t-spanner for P and a stretch factor $t \geq 1$, if for all pairs $x, y \in P$:

$$||xy|| \leq \delta_G(x, y) \leq t \cdot ||xy|| \,,$$

where $\delta_G(x, y) =$ length of the shortest x-to-y path in G.

WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and $\text{rep}(u) = x, \text{rep}(v) = y\}$.

WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and $\text{rep}(u) = x, \text{rep}(v) = y\}$.

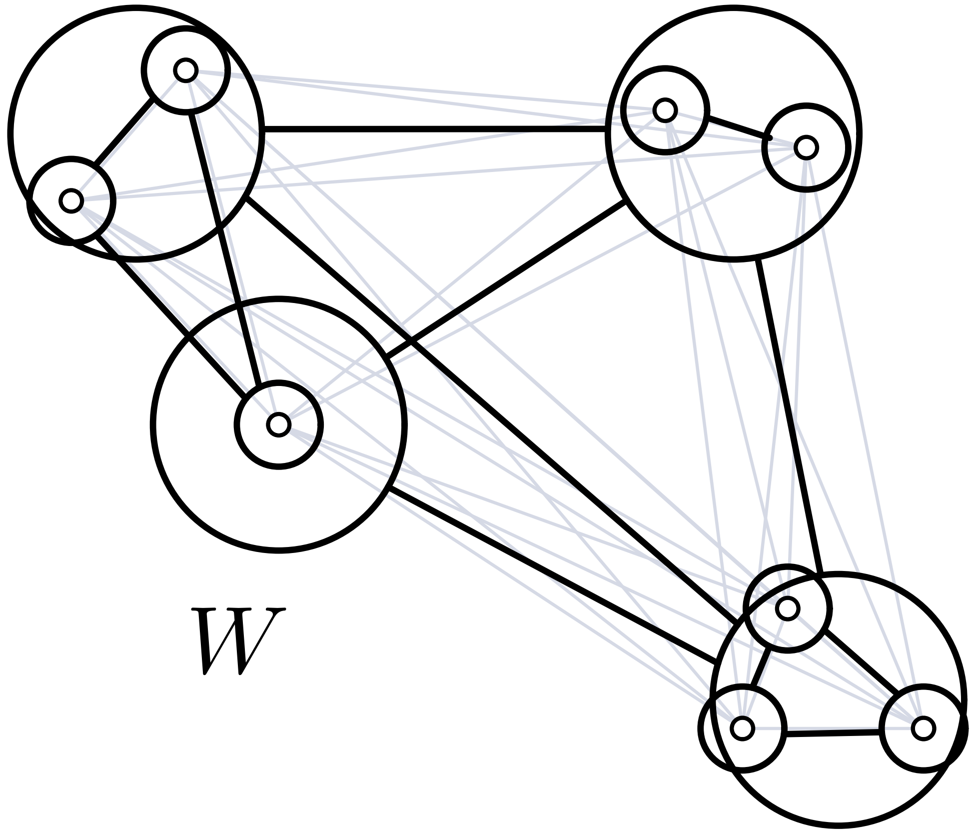
Reminder: every pair $\{u, v\} \in W$ corresponds to two quadtree nodes u and v. From each quadtree node a representative is selected in the following way. For leaf u define as representative

$$\text{rep}(u) = \begin{cases} p & \text{if } P_u = \{p\} \text{ (}u\text{ is leaf)} \\ \varnothing & \text{otherwise.} \end{cases}$$

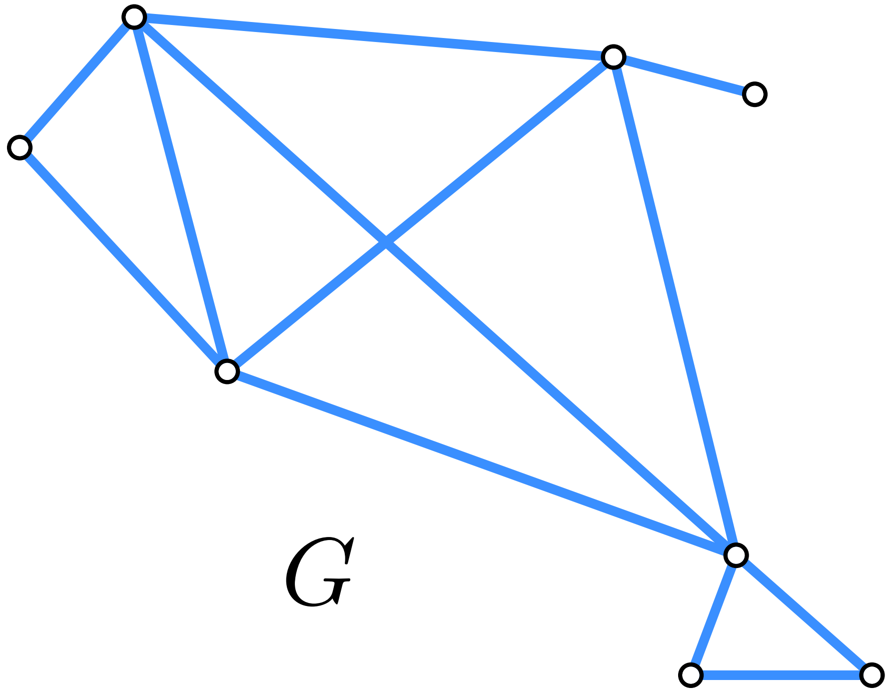
For an inner node v set $\text{rep}(v) = \text{rep}(u)$ for a non-empty child u of v.

WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and $\mathrm{rep}(u) = x, \mathrm{rep}(v) = y\}$.



W



G

WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and $\text{rep}(u) = x, \text{rep}(v) = y\}$.

Lemma: If W is an s-WSPD for $s = 4 \cdot \frac{t+1}{t-1}$, then G is a t-spanner for P with $O(s^d n)$ edges.

WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and $\text{rep}(u) = x, \text{rep}(v) = y\}$.

Lemma: If W is an s-WSPD for $s = 4 \cdot \frac{t+1}{t-1}$, then G is a t-spanner for P with $O(s^d n)$ edges.

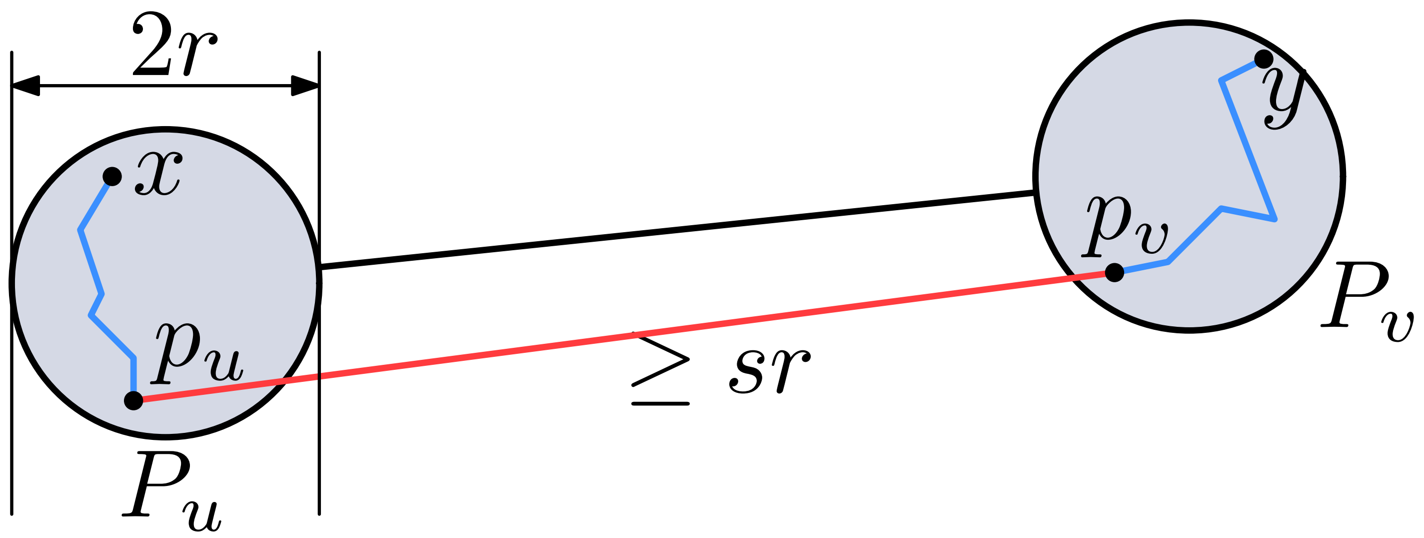
Proof: induction on distances

WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and $\mathrm{rep}(u) = x, \mathrm{rep}(v) = y\}$.

Lemma: If W is an s-WSPD for $s = 4 \cdot \frac{t+1}{t-1}$, then G is a t-spanner for P with $O(s^d n)$ edges.

Proof: induction on distances



WSPD and t-Spanner

Definition: For n points P in $\mathbb{R}^d$ and a WSPD W of P define the graph $G = (P, E)$ with $E = \{\{x, y\} \mid \{u, v\} \in W$ and rep$(u) = x$, rep$(v) = y\}$.

Lemma: If W is an s-WSPD for $s = 4 \cdot \frac{t+1}{t-1}$, then G is a t-spanner for P with $O(s^d n)$ edges.

Question: How large does s need to be if $t = 1 + \varepsilon$

A: 4

B: $O(1/\varepsilon)$

C: $O(1/\varepsilon^d)$

Summary

Theorem: For a set P of n points in $\mathbb{R}^d$ and an $\varepsilon \in (0, 1]$ a $(1 + \varepsilon)$-spanner for P with $O(n/\varepsilon^d)$ edges can be computed in $O(n \log n + n/\varepsilon^d)$ time.

Summary

Theorem: For a set P of n points in $\mathbb{R}^d$ and an $\varepsilon \in (0,1]$ a $(1+\varepsilon)$-spanner for P with $O(n/\varepsilon^d)$ edges can be computed in $O(n \log n + n/\varepsilon^d)$ time.

Proof: For $t = (1+\varepsilon)$ it holds with $s = 4 \cdot \frac{t+1}{t-1}$

$$O(s^d n) = O\left(\left(4 \cdot \frac{2+\varepsilon}{\varepsilon}\right)^d n\right) \subseteq O\left(\left(\frac{12}{\varepsilon}\right)^d n\right) =$$

Summary

Theorem: For a set P of n points in $\mathbb{R}^d$ and an $\varepsilon \in (0,1]$ a $(1+\varepsilon)$-spanner for P with $O(n/\varepsilon^d)$ edges can be computed in $O(n \log n + n/\varepsilon^d)$ time.

Proof: For $t = (1+\varepsilon)$ it holds with $s = 4 \cdot \frac{t+1}{t-1}$

$$O(s^d n) = O\left(\left(4 \cdot \frac{2+\varepsilon}{\varepsilon}\right)^d n\right) \subseteq O\left(\left(\frac{12}{\varepsilon}\right)^d n\right) =$$

$$\begin{array}{cc}
P & \\
\downarrow & O(n \log n) \\
\text{compressed quadtree} & \\
\downarrow & O(n/\varepsilon^d) \\
\text{WSPD} & \\
\downarrow & O(n/\varepsilon^d) \\
(1+\varepsilon)\text{-spanner} &
\end{array}$$

$\square$

Discussion

Applications of the WSPD?

WSPD is always useful, when we don't need the $\Theta(n^2)$ exact distances, but approximate distances are enough

Discussion

Applications of the WSPD?

WSPD is always useful, when we don't need the $\Theta(n^2)$ exact distances, but approximate distances are enough

Can't we compute exact solutions in the same time?

Often in $\mathbb{R}^2$ yes, but not in $\mathbb{R}^d$ for $d > 2$ (EMST, diameter).
EMST, Voronoi diagrams, ... can be computed in $O(n)$ time from quadtress/WSPDs

Additional highlights in book

- very simple from WSPD: closest pair and approximate diameter
- with basic geometry from WSPD: nearest neighbor graph
- semi-separated pair decomposition