Plan for today

finish going over exam

worksheet to start on challenge problem #1

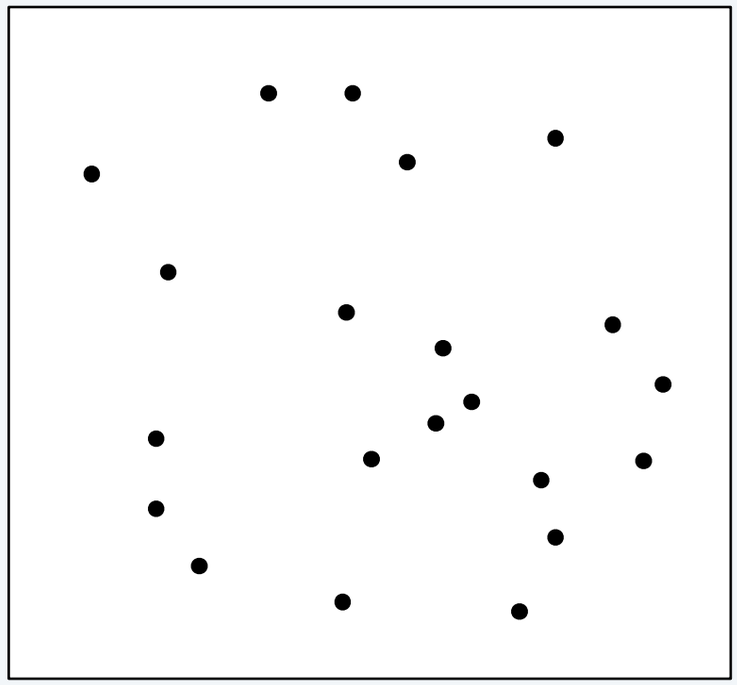
problem 1 should be EASY

dist btwn 2 points in 2D:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

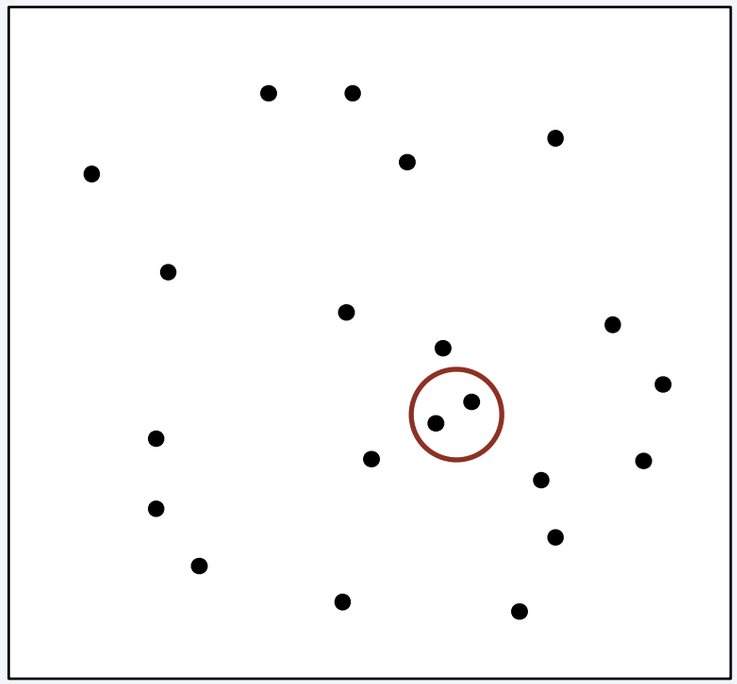
Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.



Closest pair of points

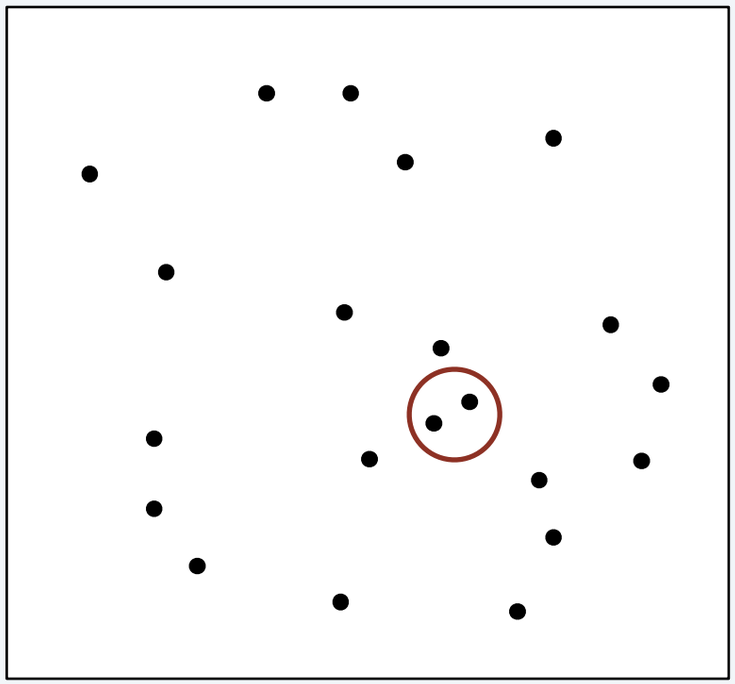
Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.



Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.

Fundamental geometric primitive.

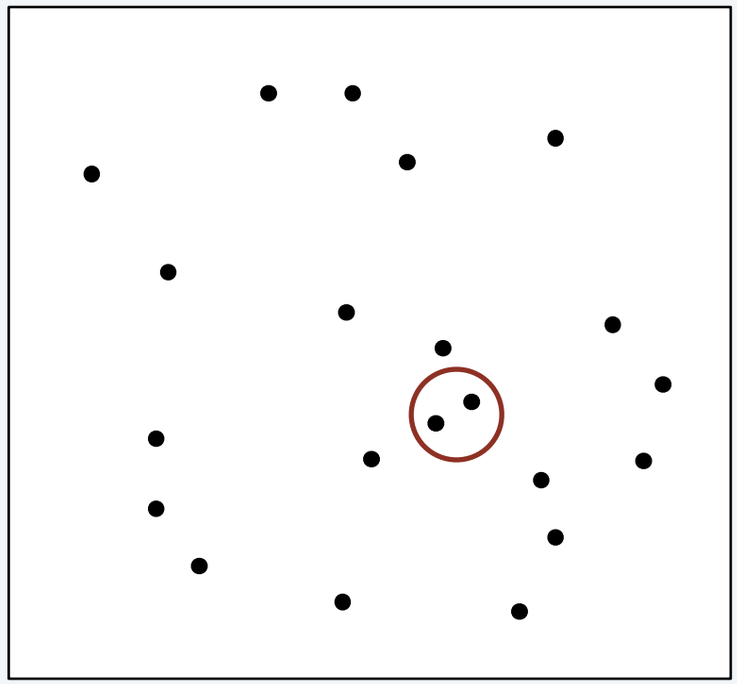


Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.

Fundamental geometric primitive.

- Graphics, computer vision, geographic information systems, molecular modeling, air traffic control.



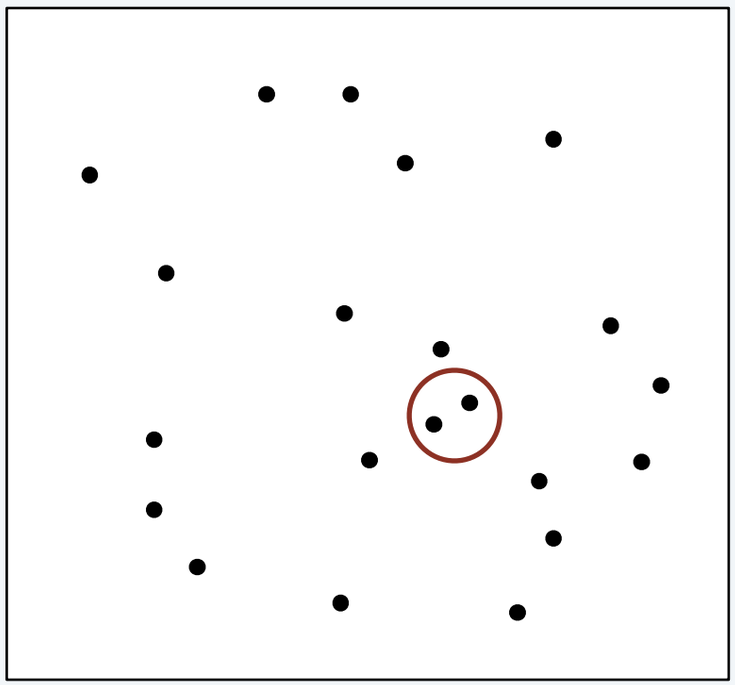
Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.

Fundamental geometric primitive.

- Graphics, computer vision, geographic information systems, molecular modeling, air traffic control.
- Special case of nearest neighbor, Euclidean MST, Voronoi.

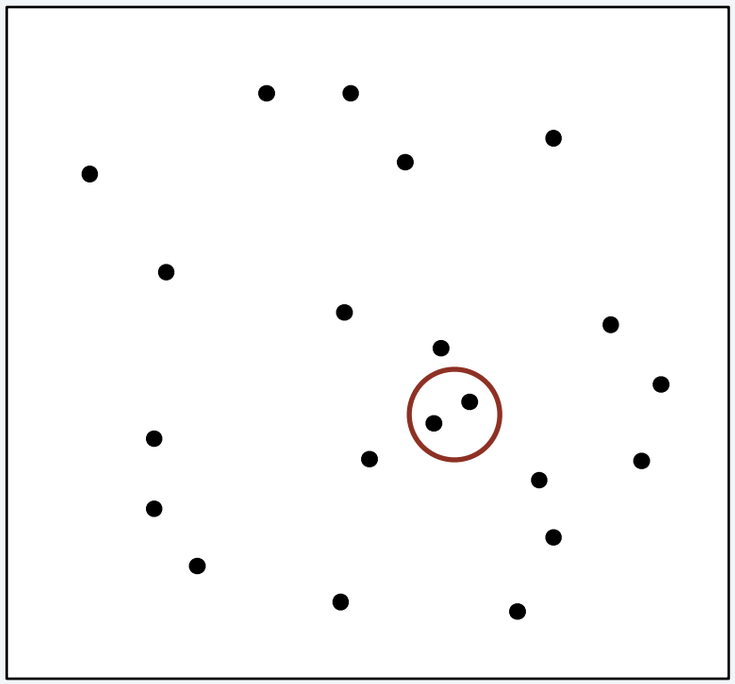
fast closest pair inspired fast algorithms for these problems



Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.

Brute force. Check all pairs with $\Theta(n^2)$ distance calculations.



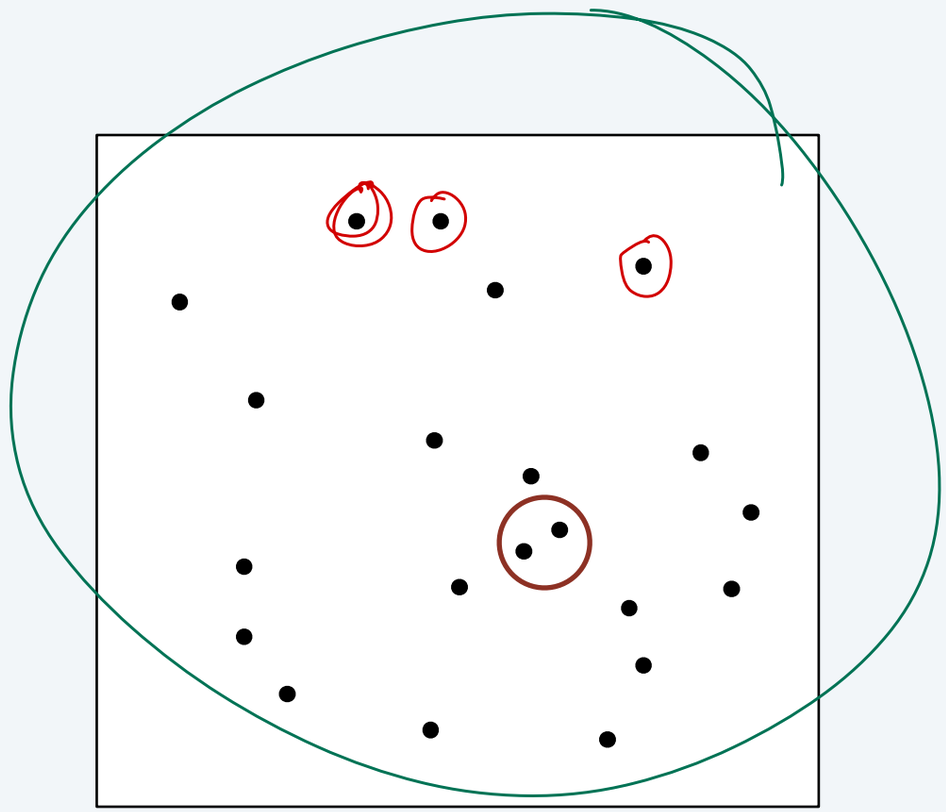
Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.

Brute force. Check all pairs with $\Theta(n^2)$ distance calculations.

1D version. Easy $O(n \log n)$ algorithm.

sort, then check adjacent pairs



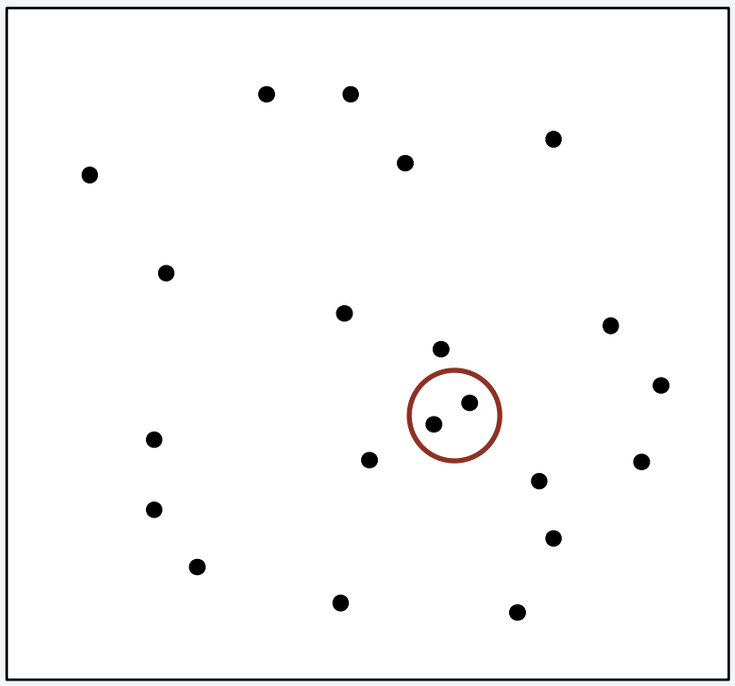
Closest pair of points

Closest pair problem. Given n points in the plane, find a pair of points with the smallest Euclidean distance between them.

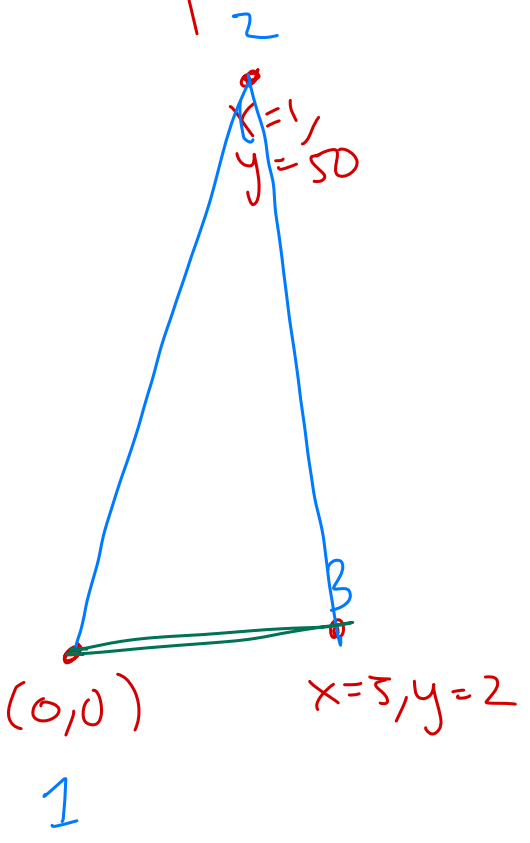
Brute force. Check all pairs with $\Theta(n^2)$ distance calculations.

1D version. Easy $O(n \log n)$ algorithm.

Non-degeneracy assumption. No two points have the same x-coordinate.



sort by x coordinate
compare adjacent points



Remember merge sort?

```
mergesort(L):
  L_1 = first half of L
  L_2 = first half of L
  sorted_L_1 = mergesort(L_1)
  sorted_L_2 = mergesort(L_2)
  return merged L_1 and L_2
```

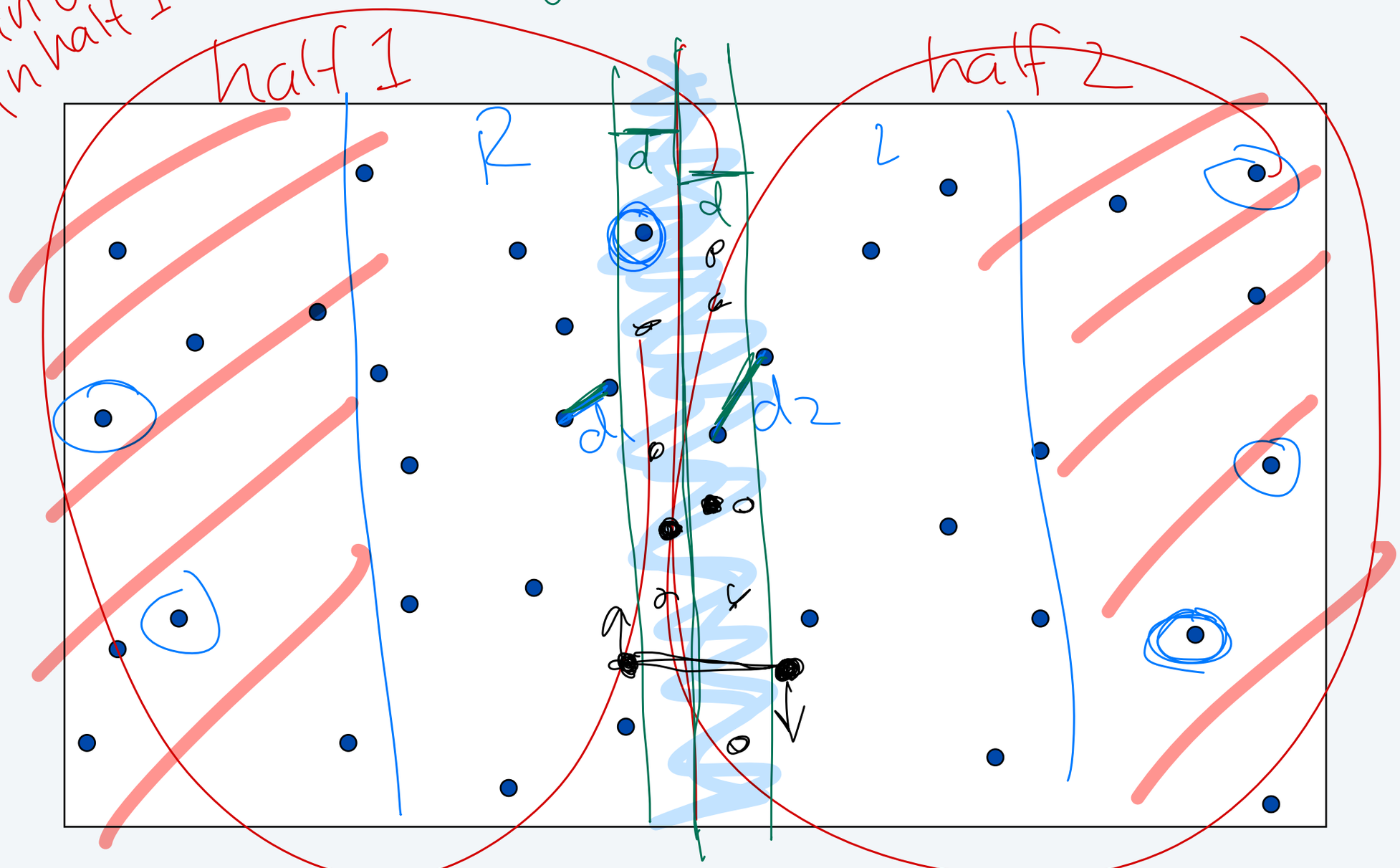
Closest pair of points: divide-and-conquer algorithm

we only need the closest points where one comes from H1 and one from H2 if they are closer than the closest points in H1 and in H2.

$d = \min(d_1, d_2)$

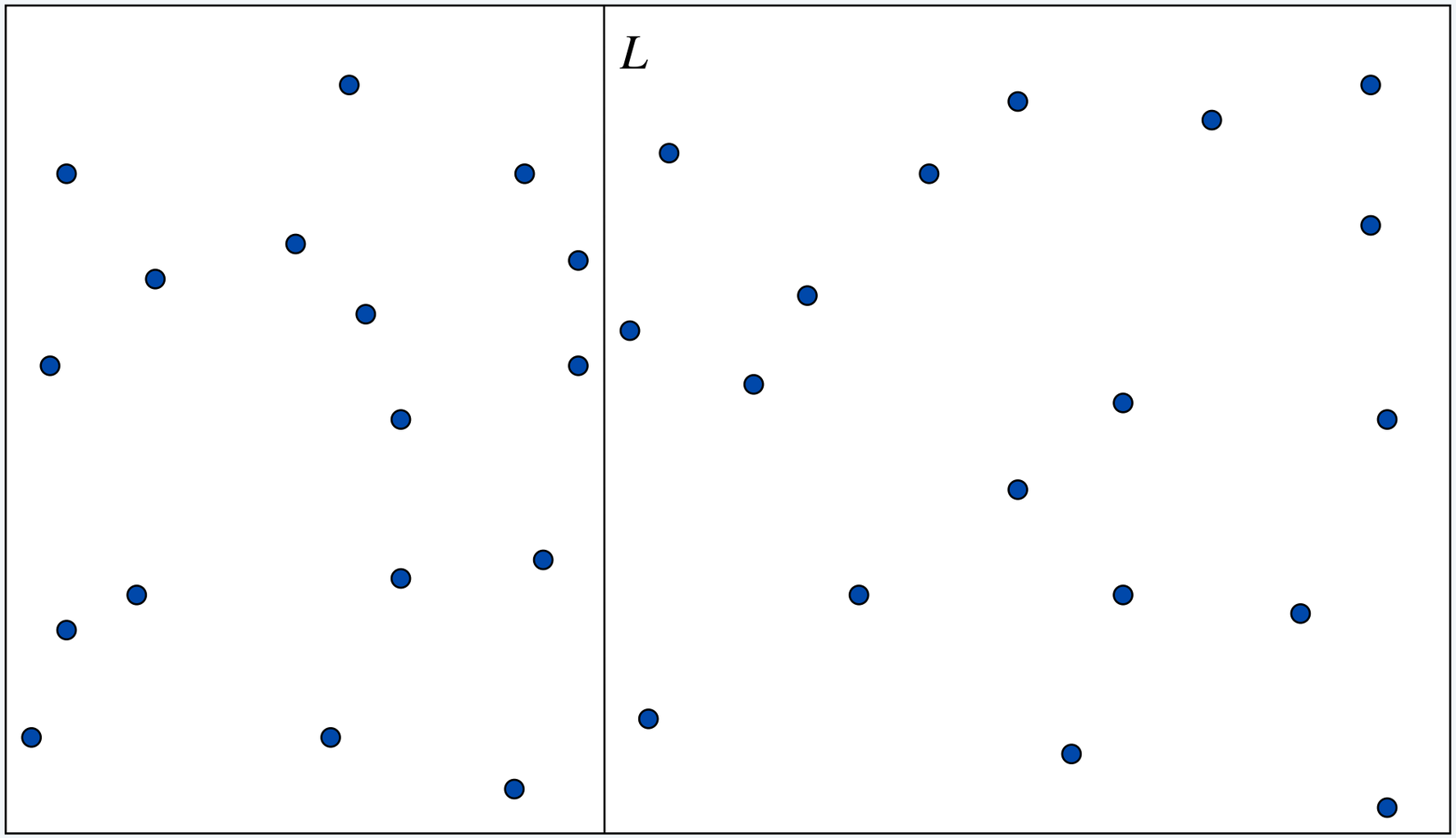
min dist in half 1

min dist half 2



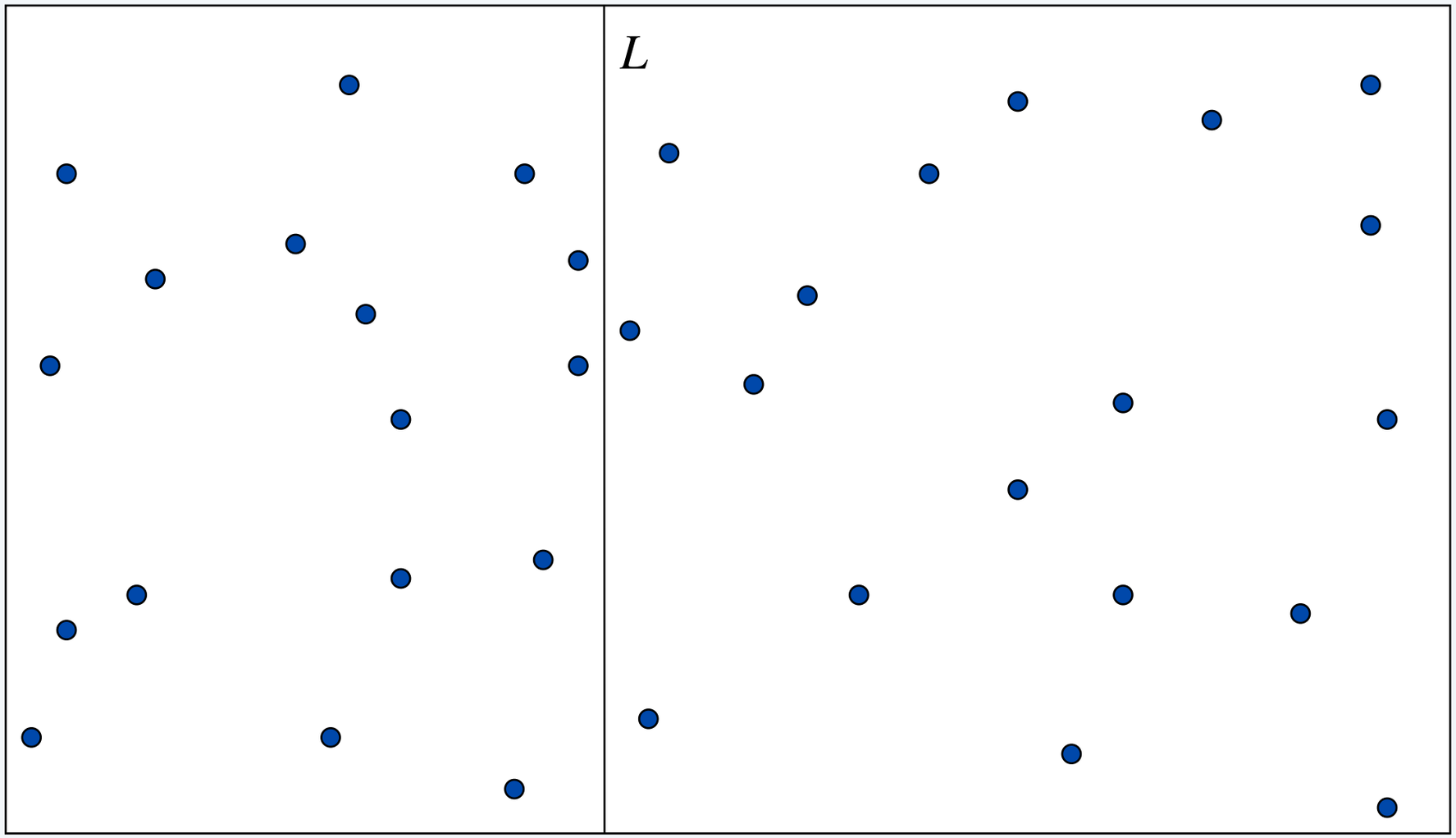
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.



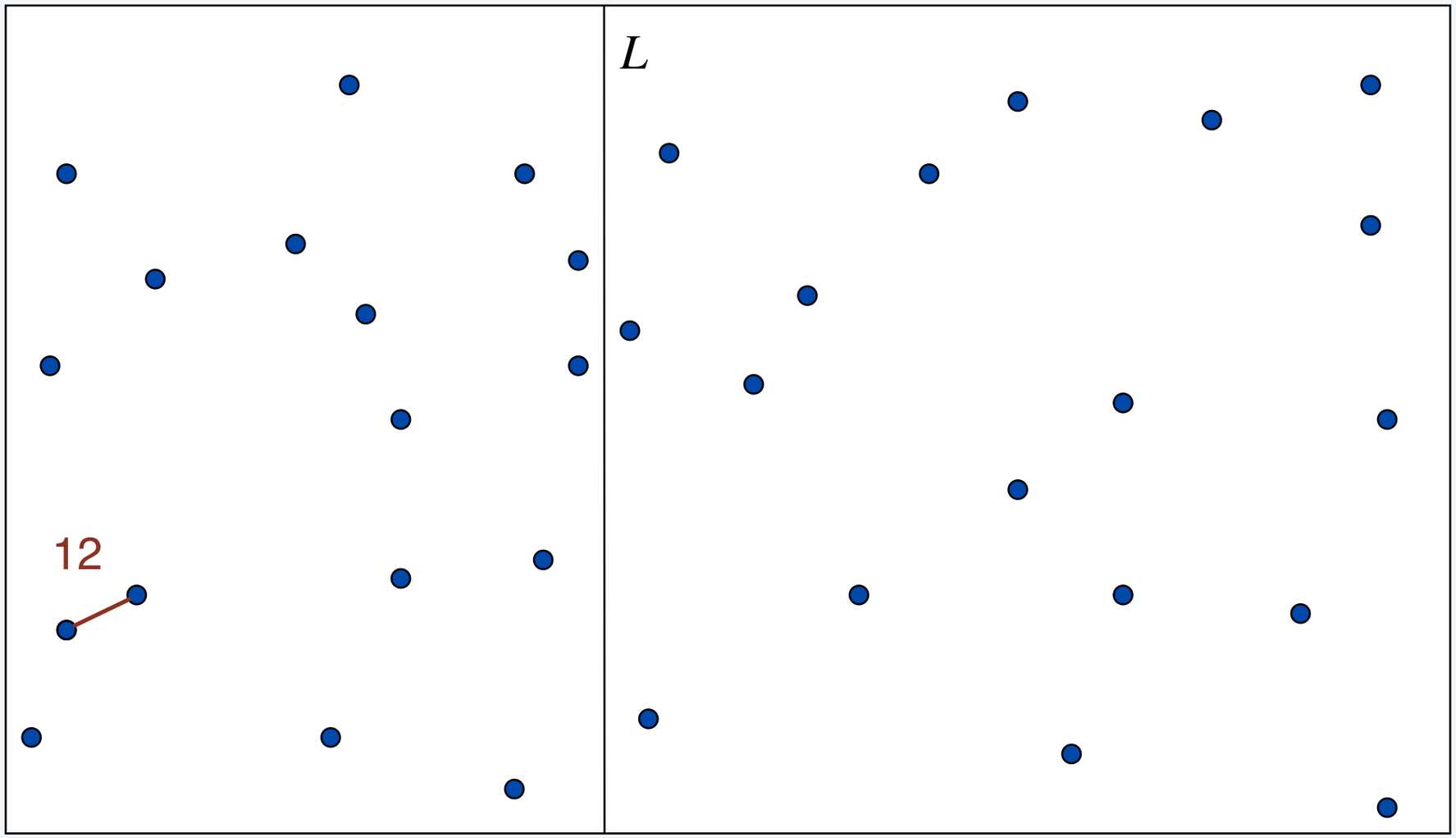
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.



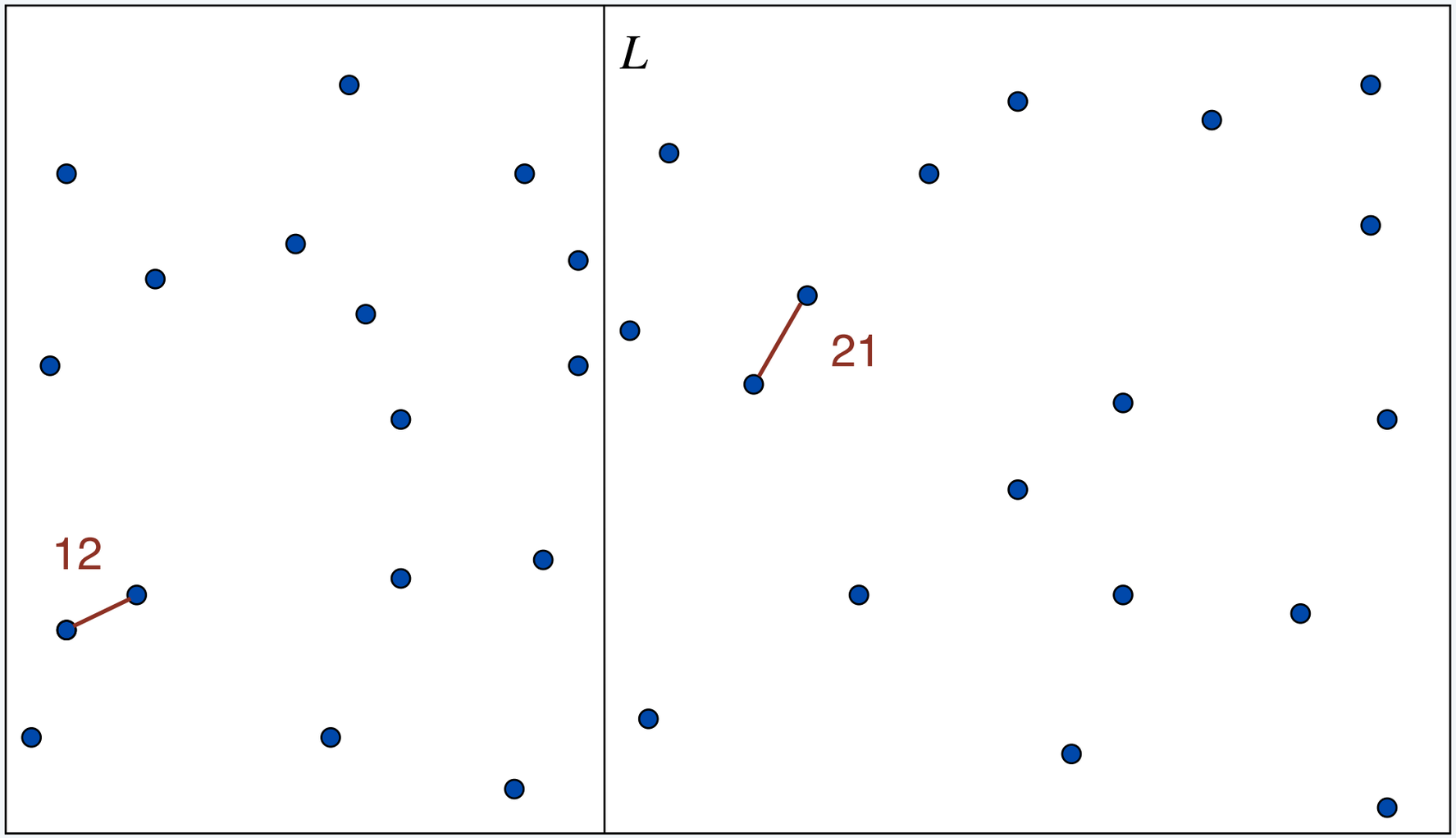
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.



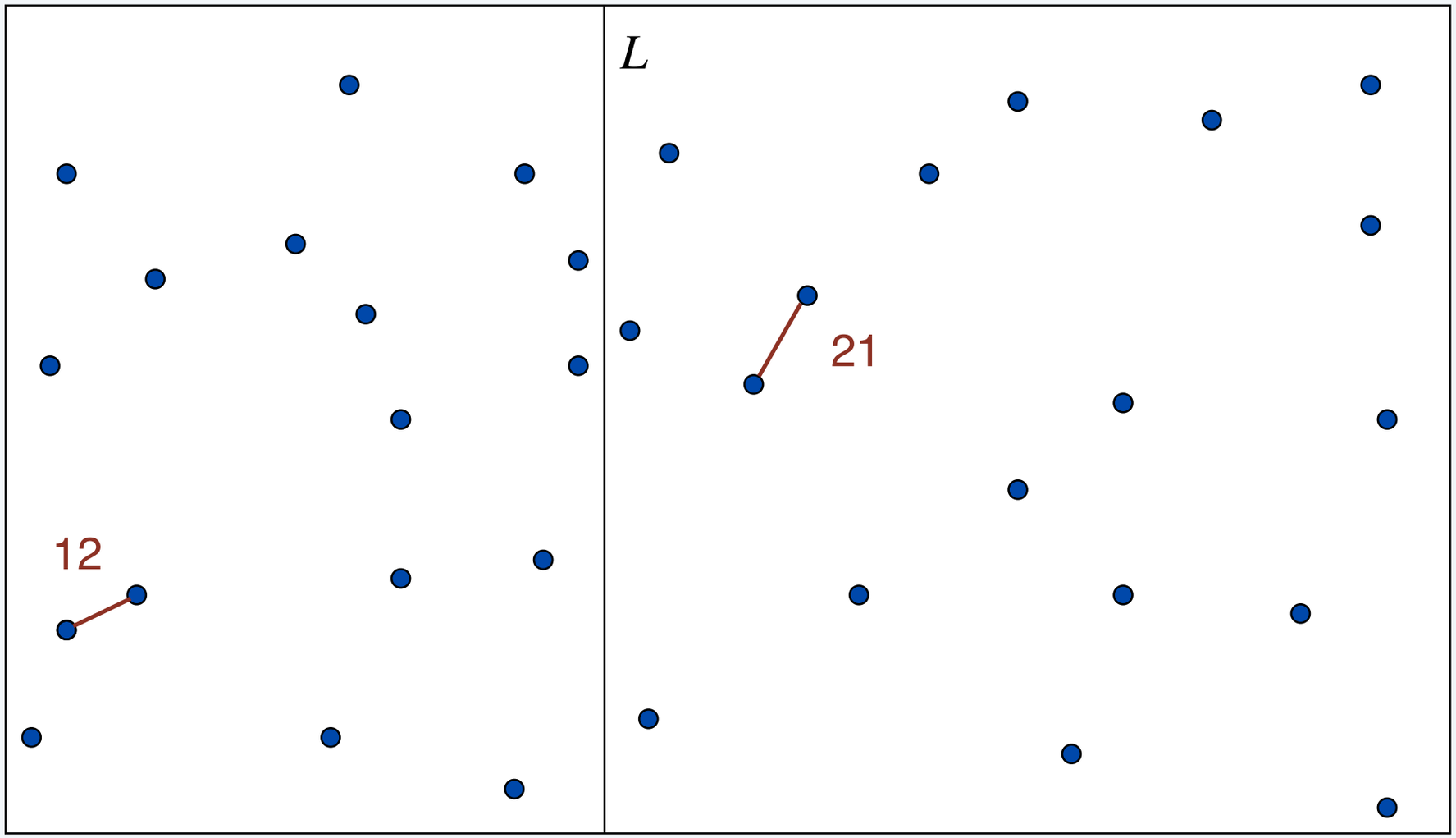
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.



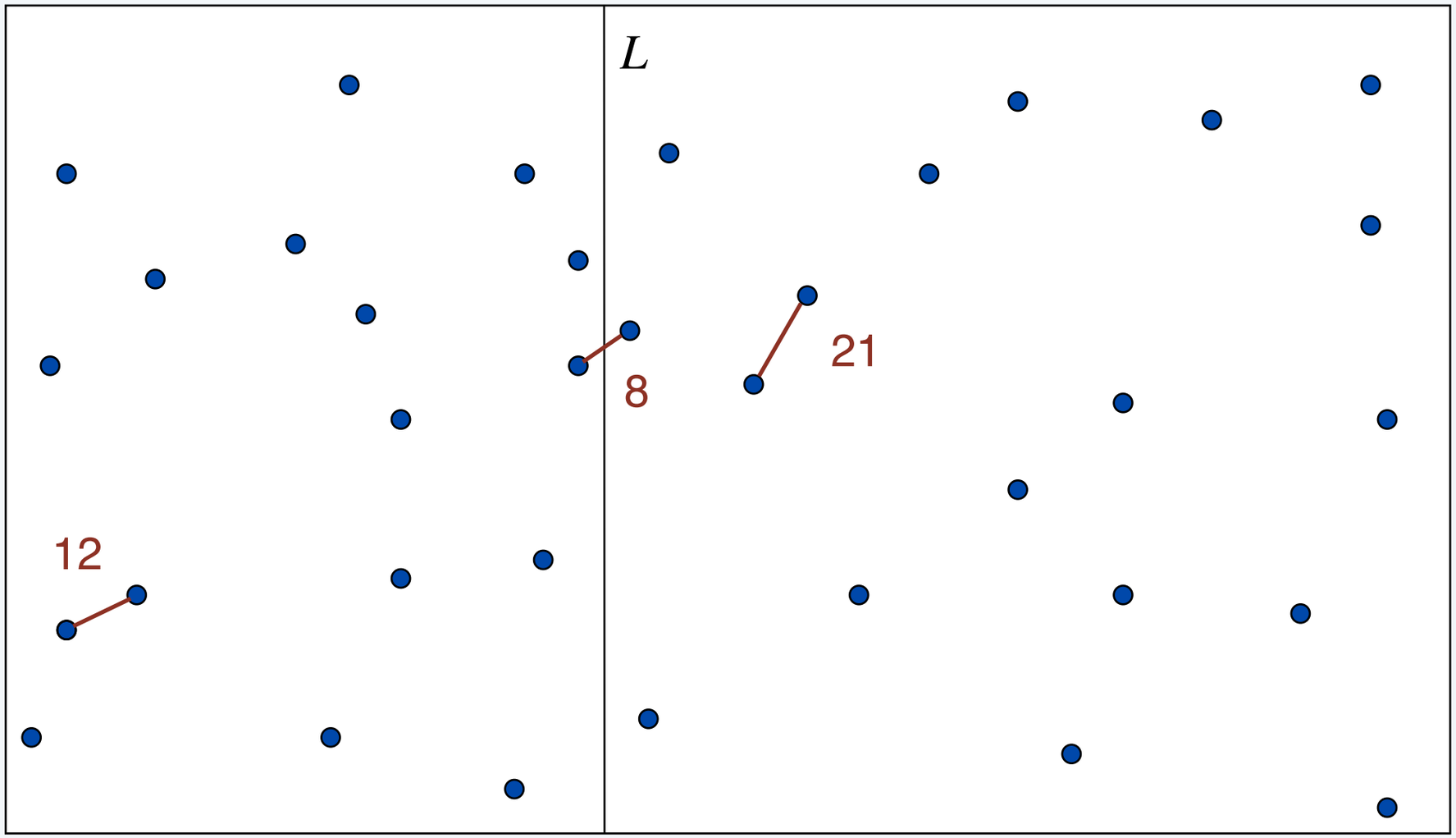
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.
- Combine: find closest pair with one point in each side.



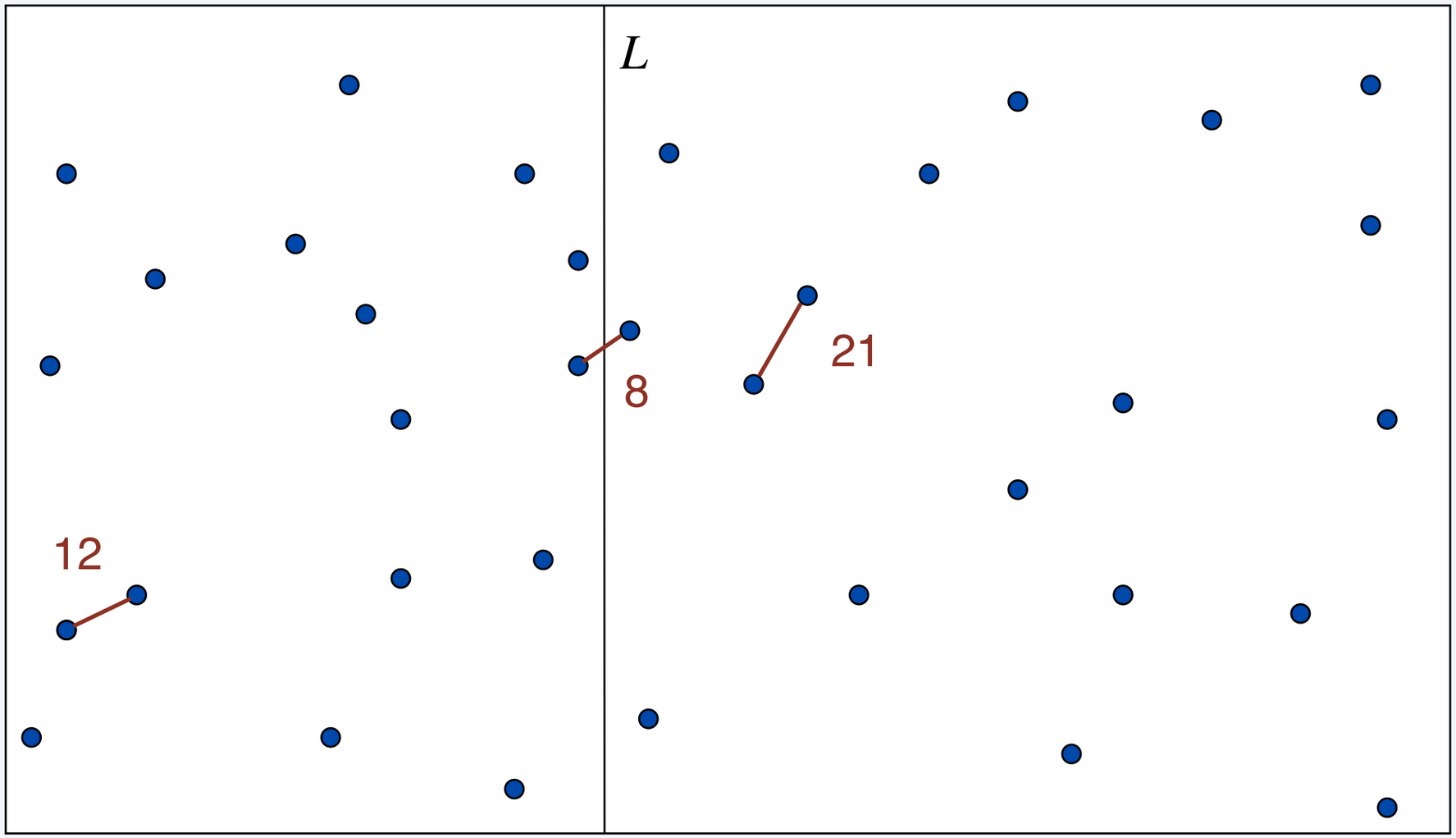
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.
- Combine: find closest pair with one point in each side.



Closest pair of points: divide-and-conquer algorithm

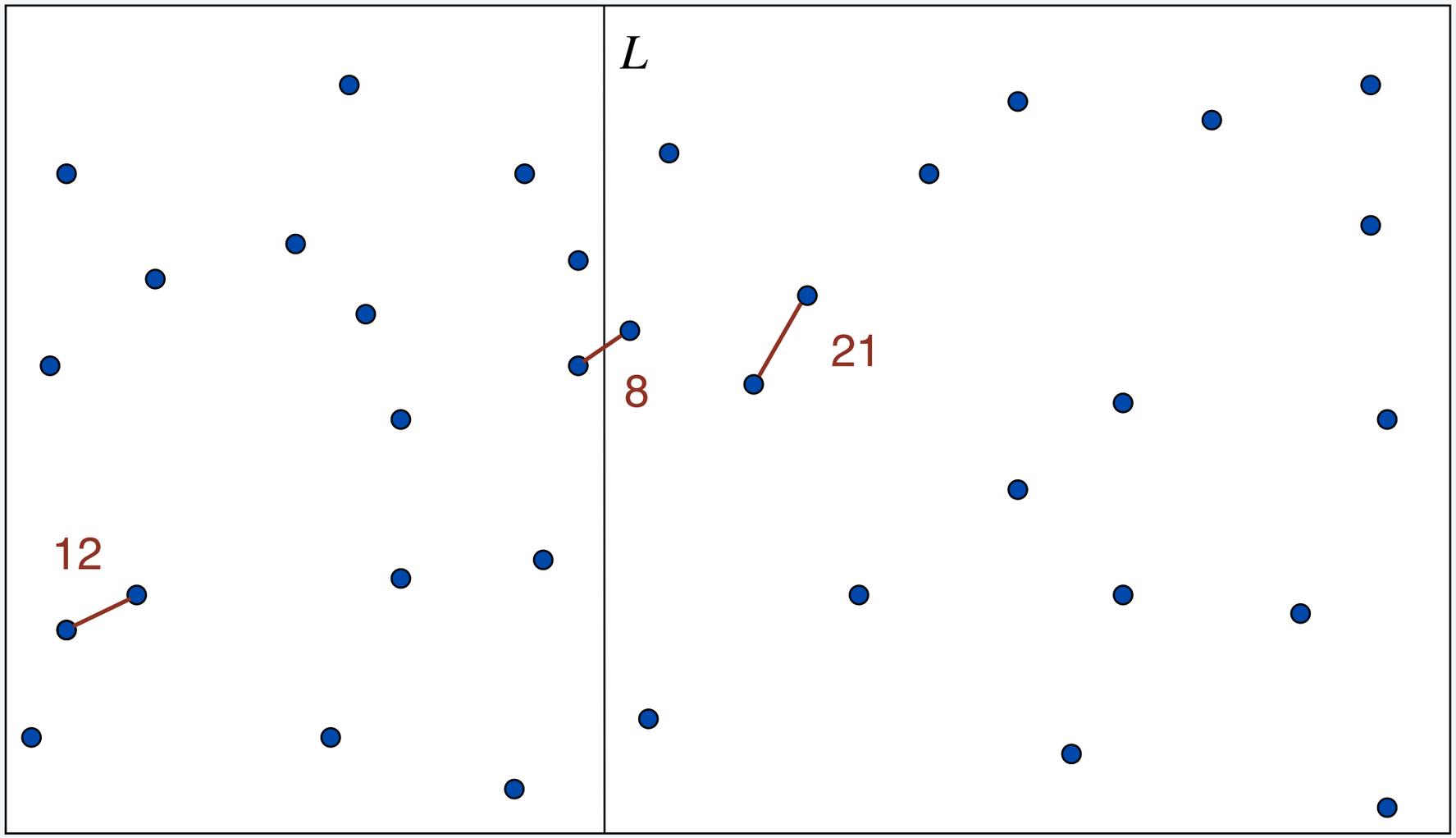
- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.
- Combine: find closest pair with one point in each side.
- Return best of 3 solutions.



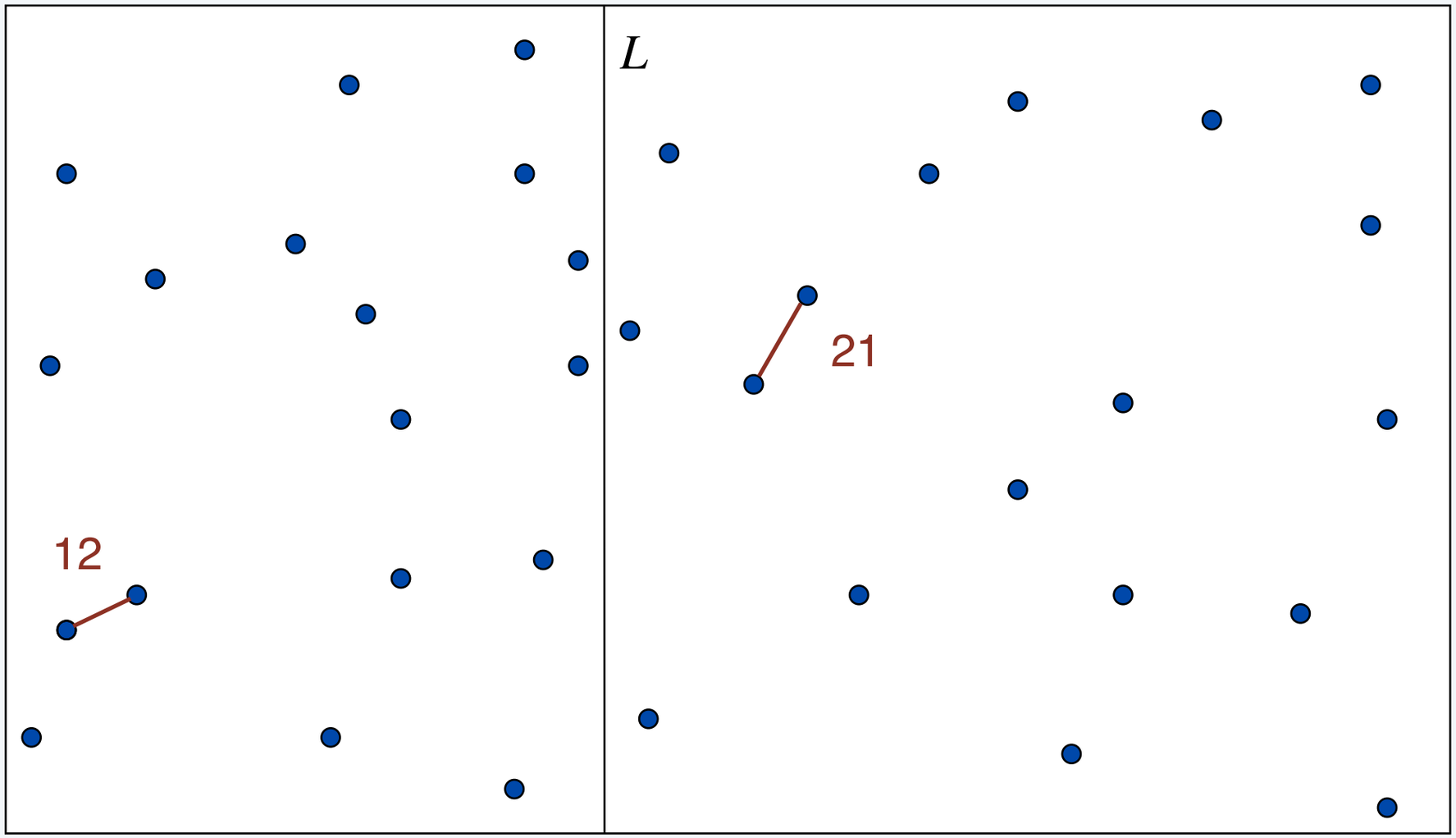
Closest pair of points: divide-and-conquer algorithm

- Divide: draw vertical line L so that $n/2$ points on each side.
- Conquer: find closest pair in each side recursively.
- Combine: find closest pair with one point in each side.
- Return best of 3 solutions.

seems like $\Theta(n^2)$

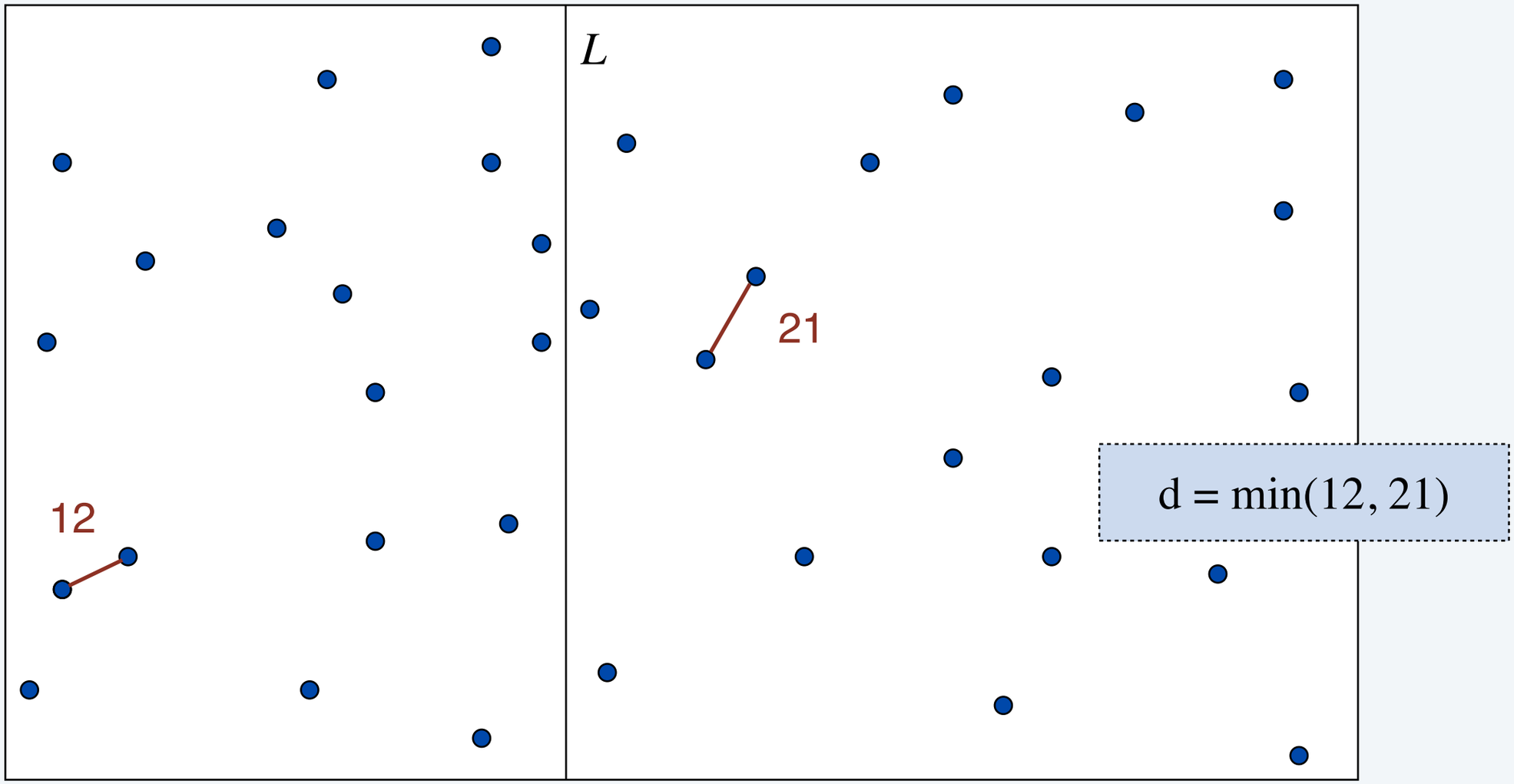


How to find closest pair with one point in each side?



How to find closest pair with one point in each side?

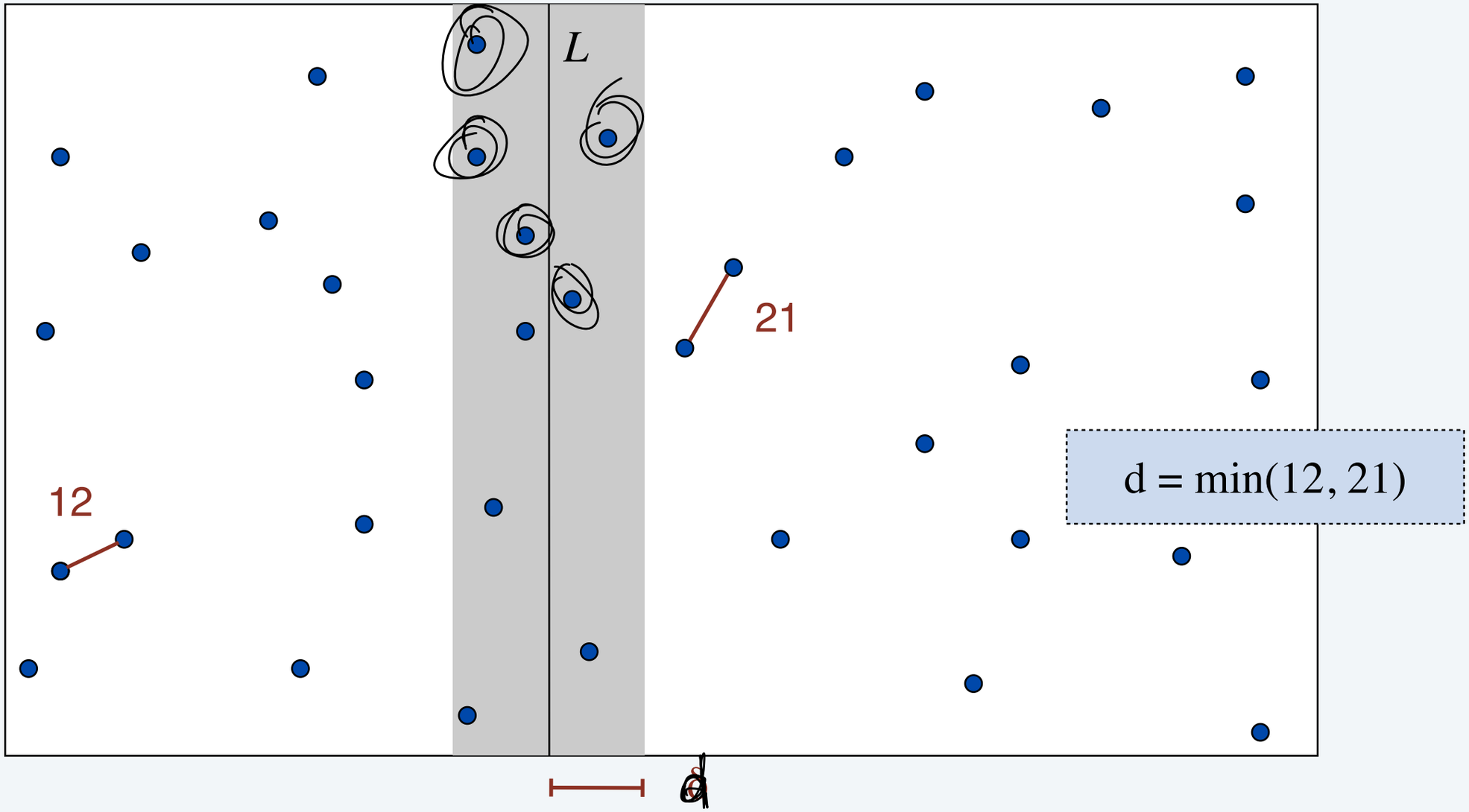
Find closest pair with one point in each side, assuming that distance < d.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

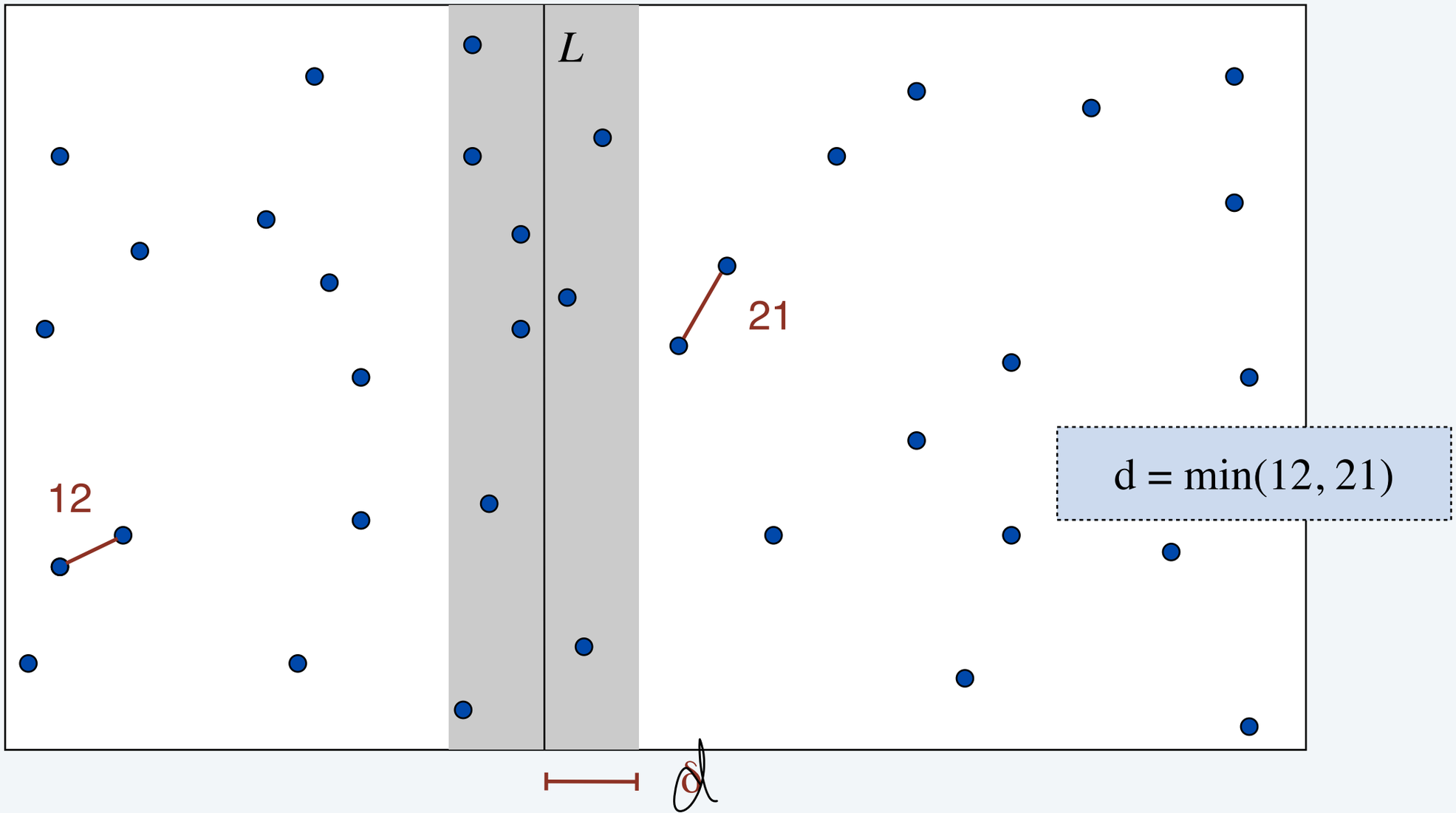
- Observation: suffices to consider only those points within d of line L.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

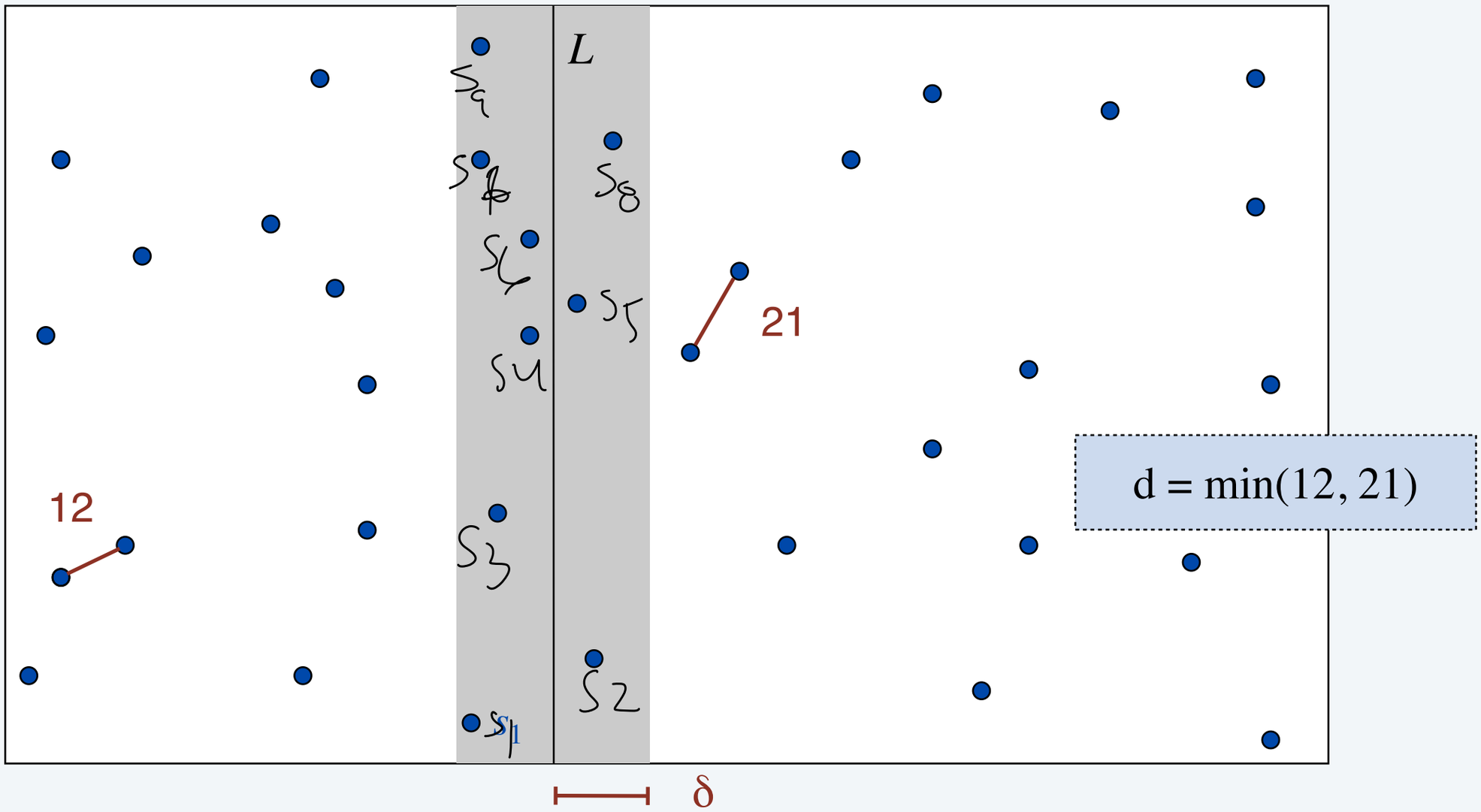
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

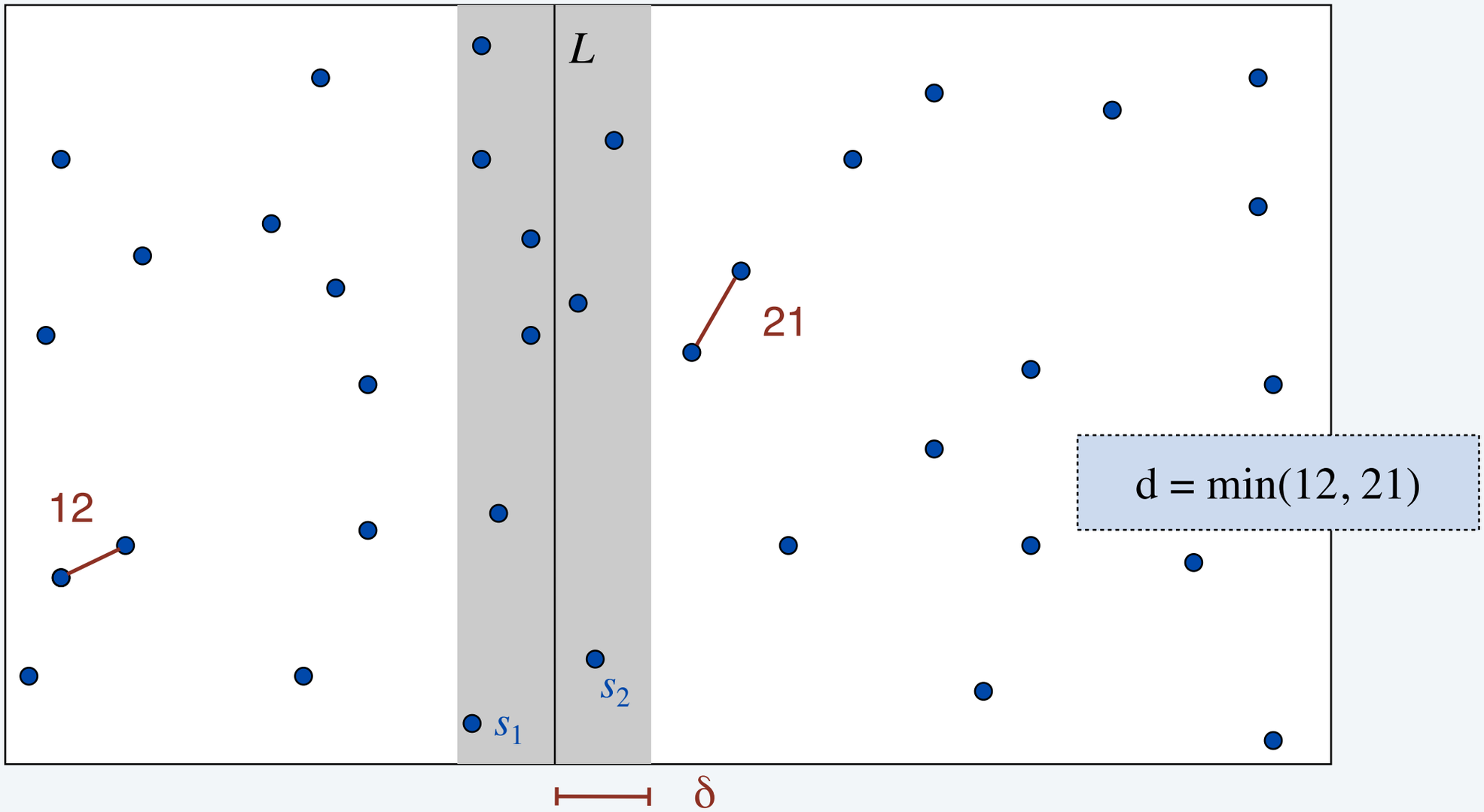
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

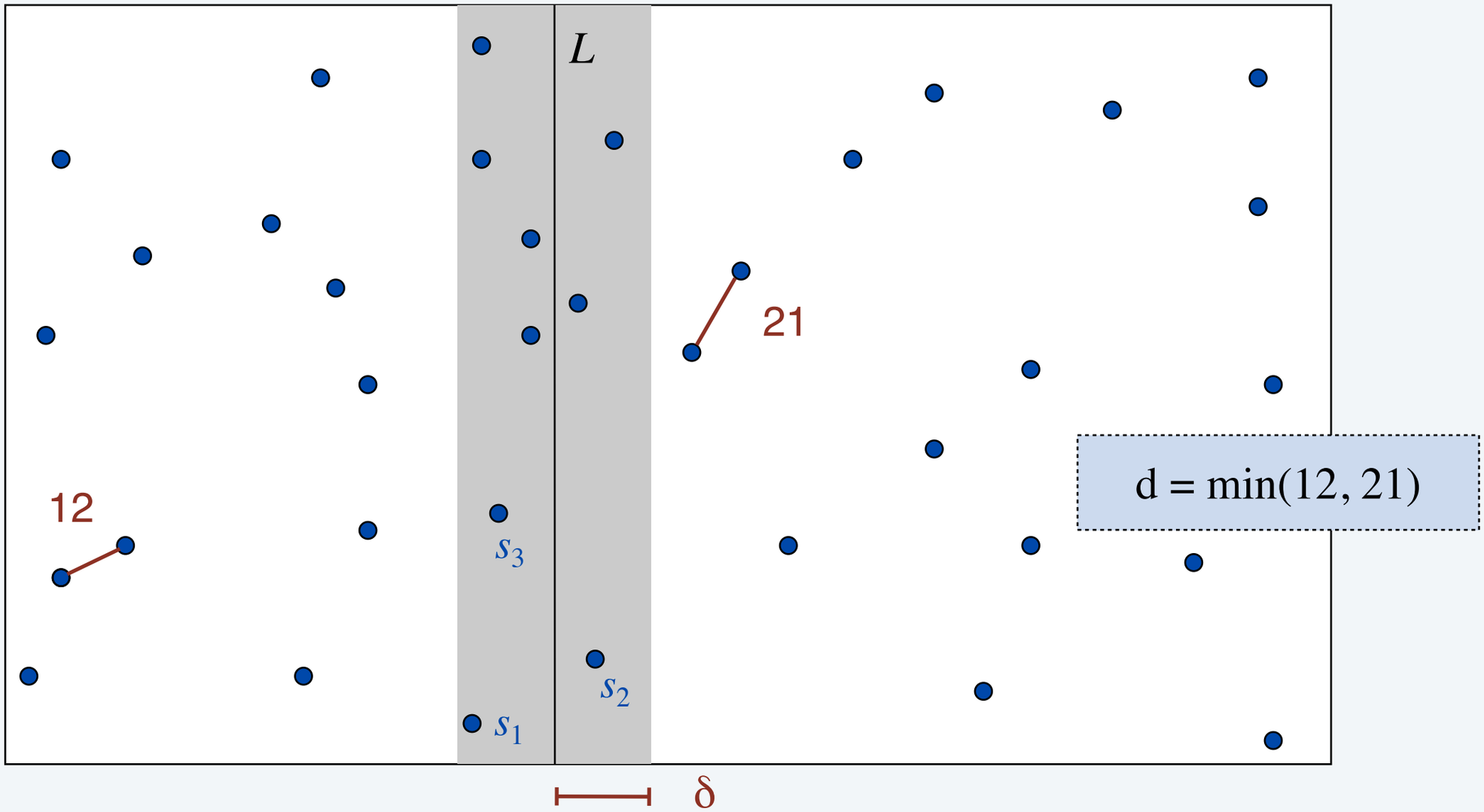
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

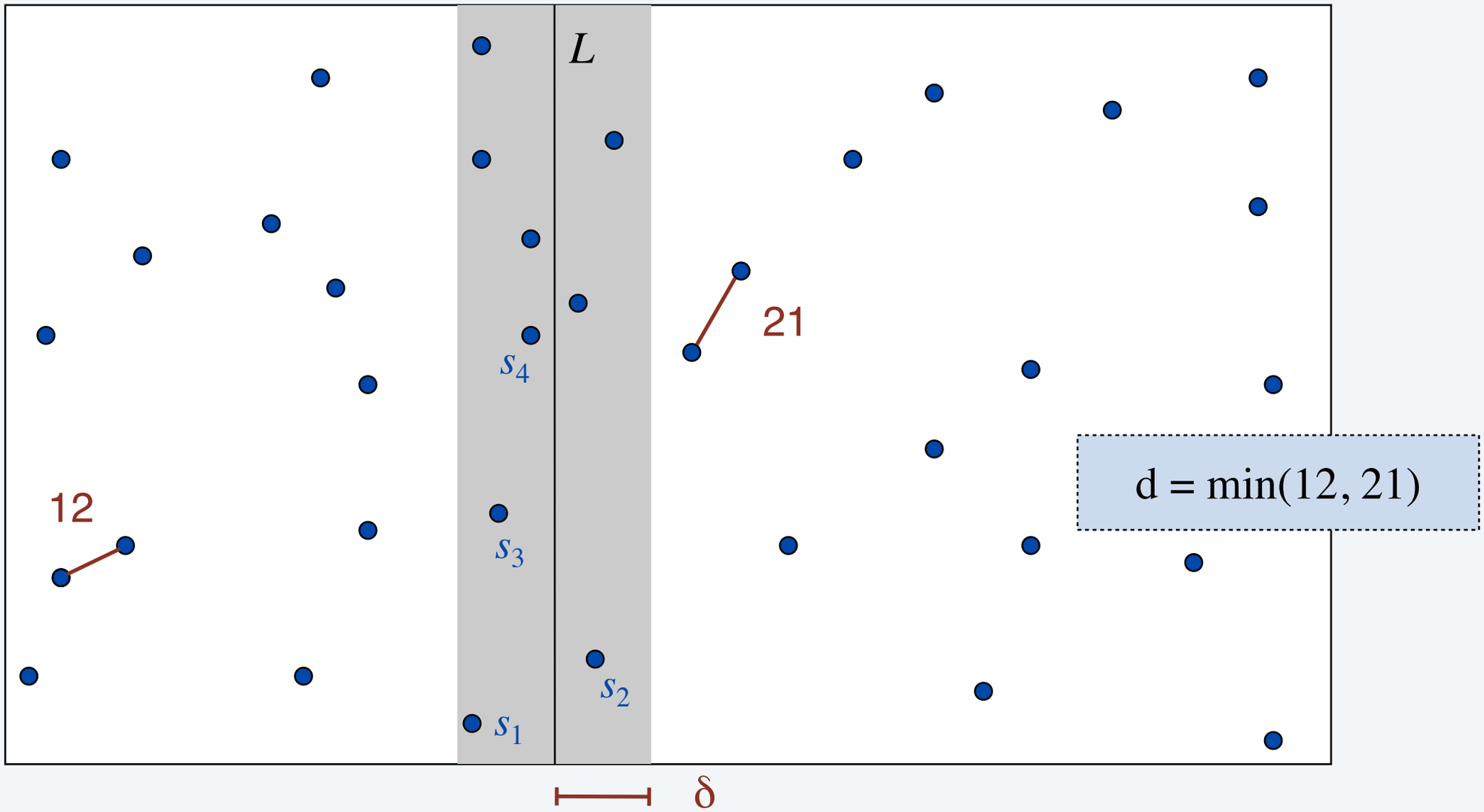
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

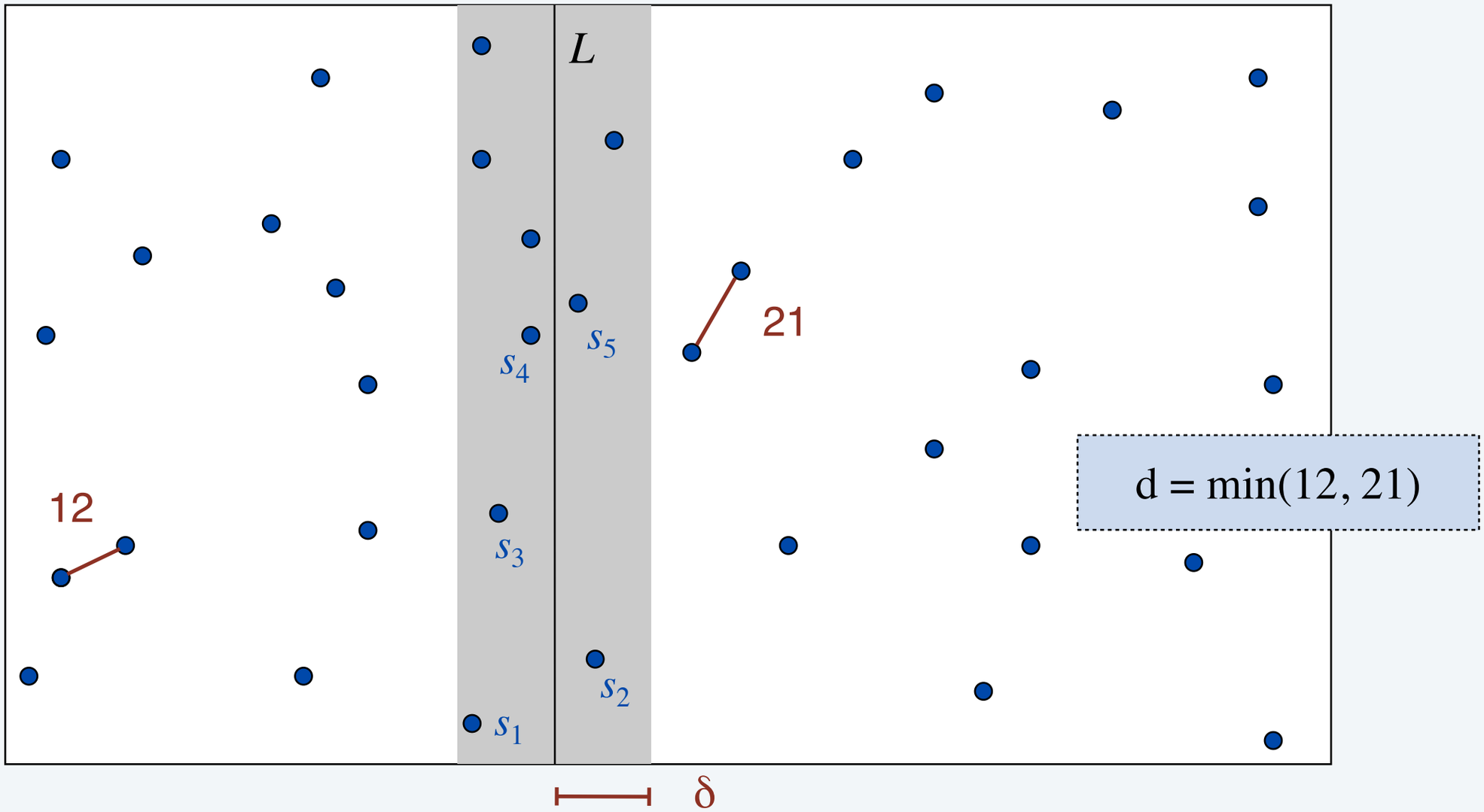
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

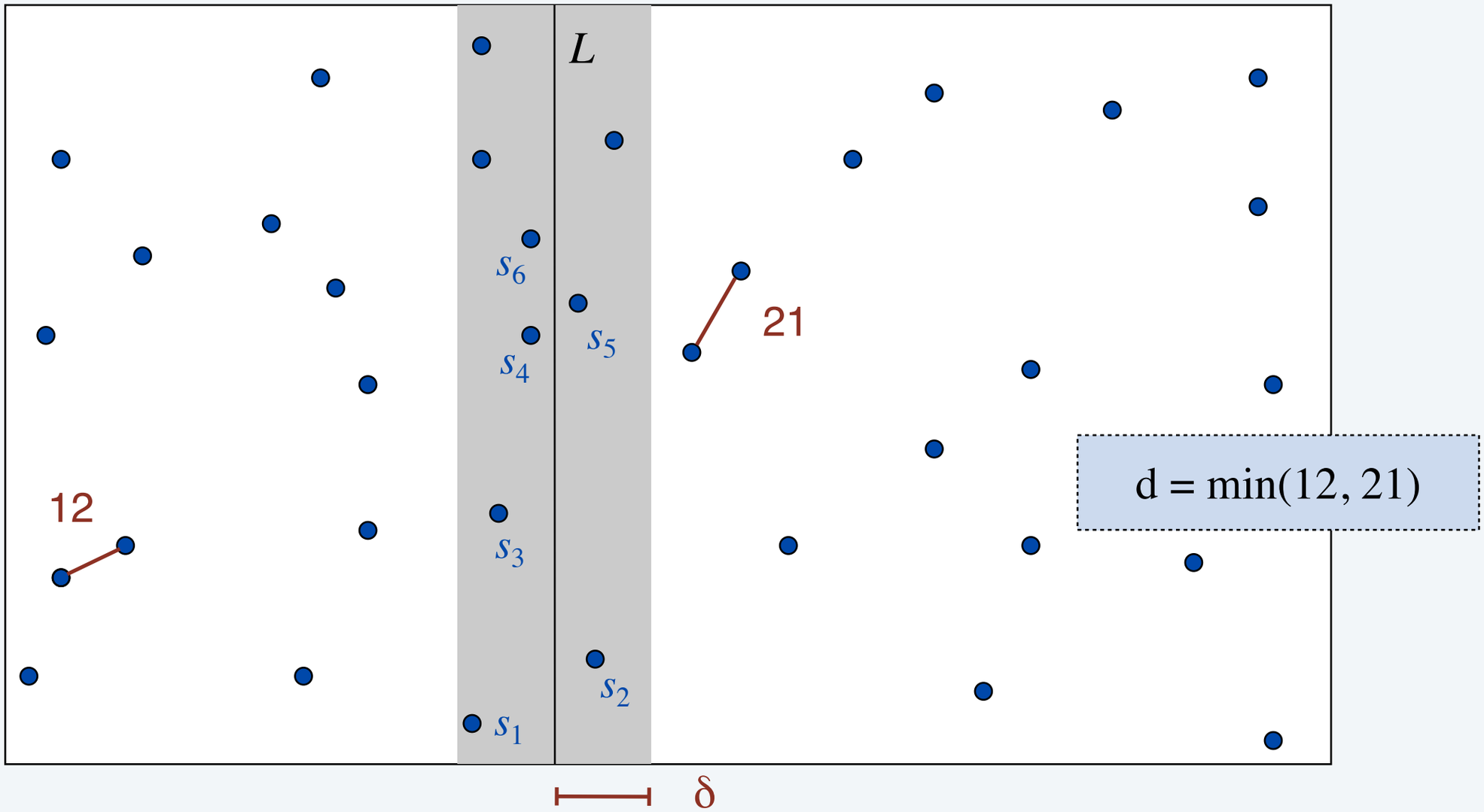
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

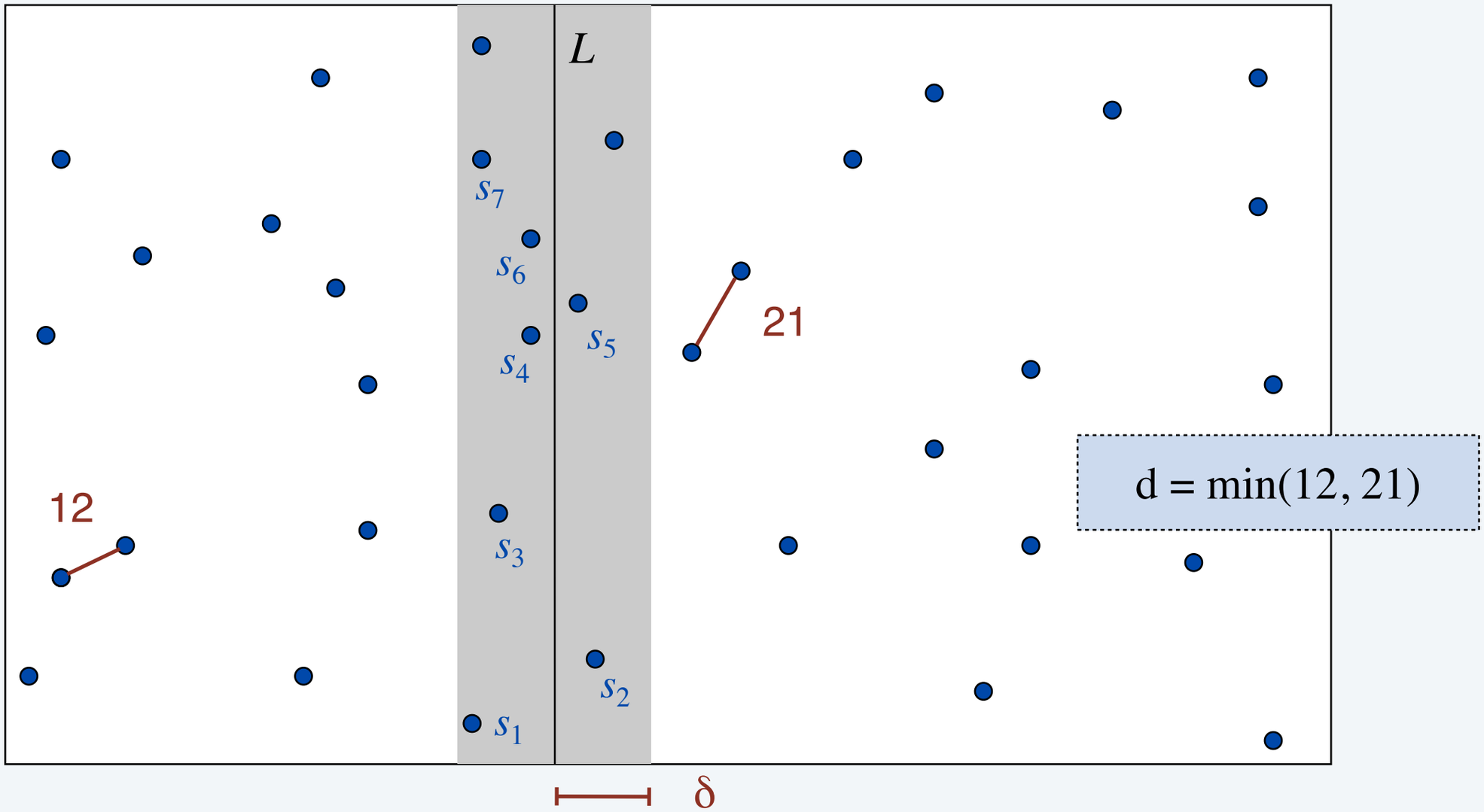
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

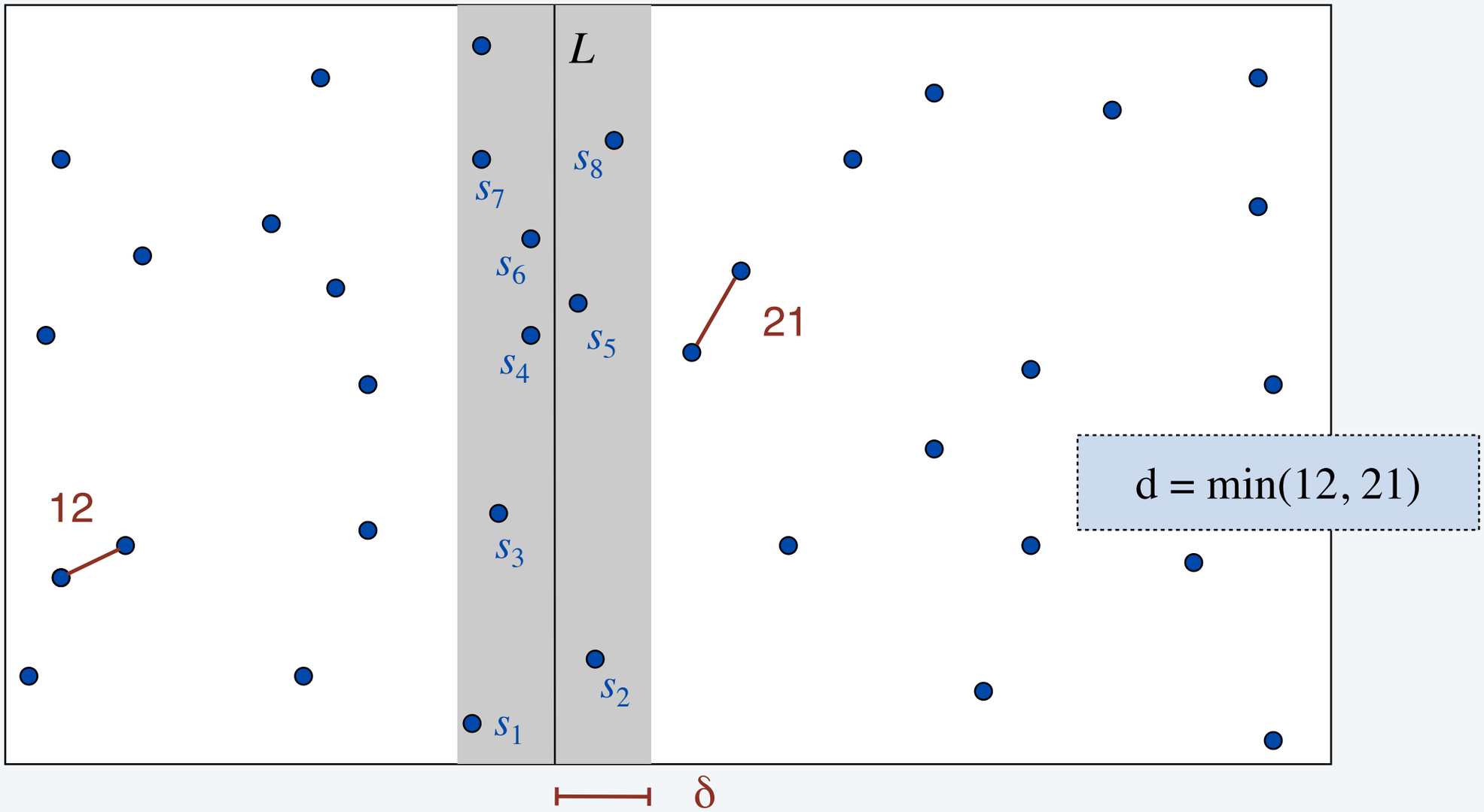
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

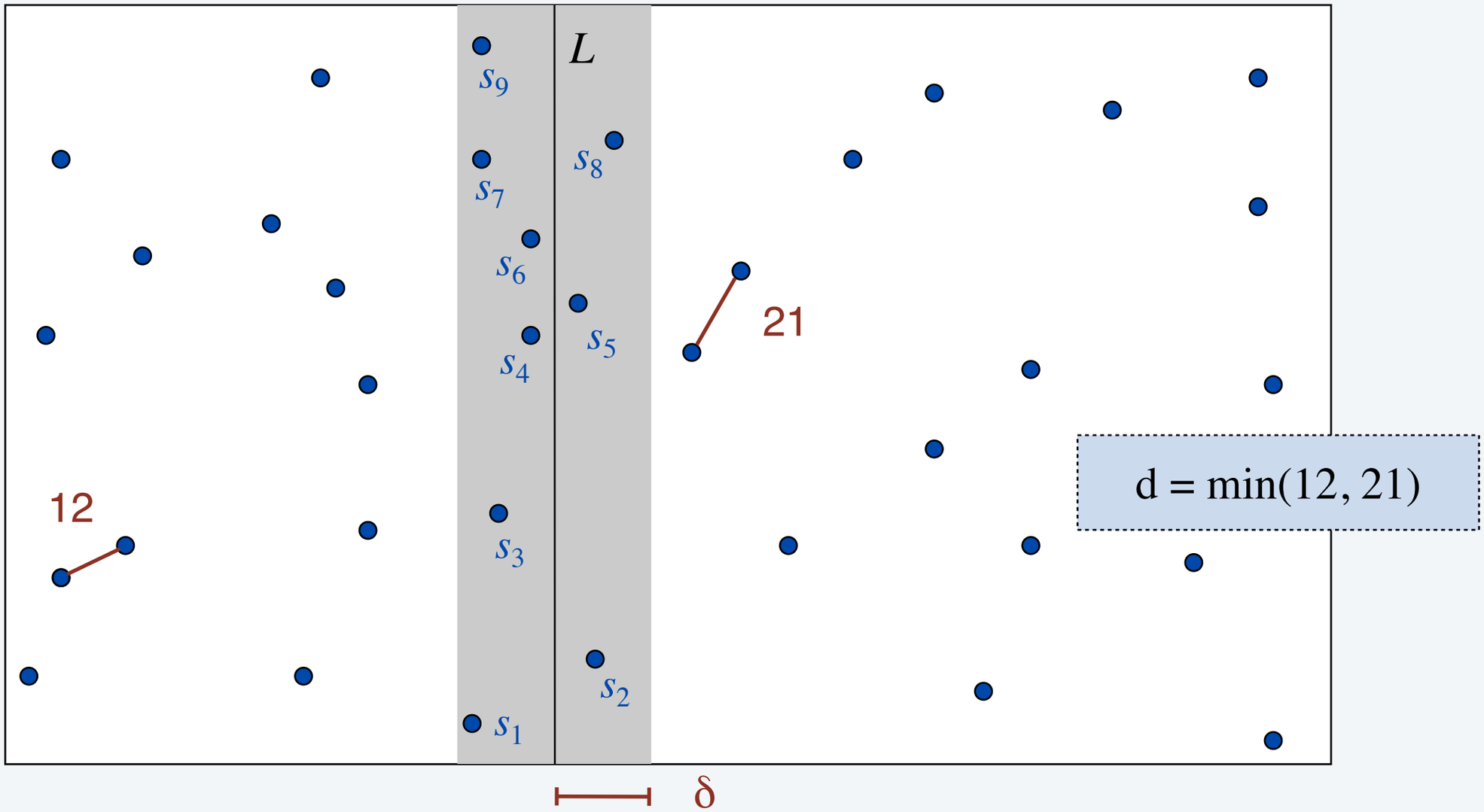
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

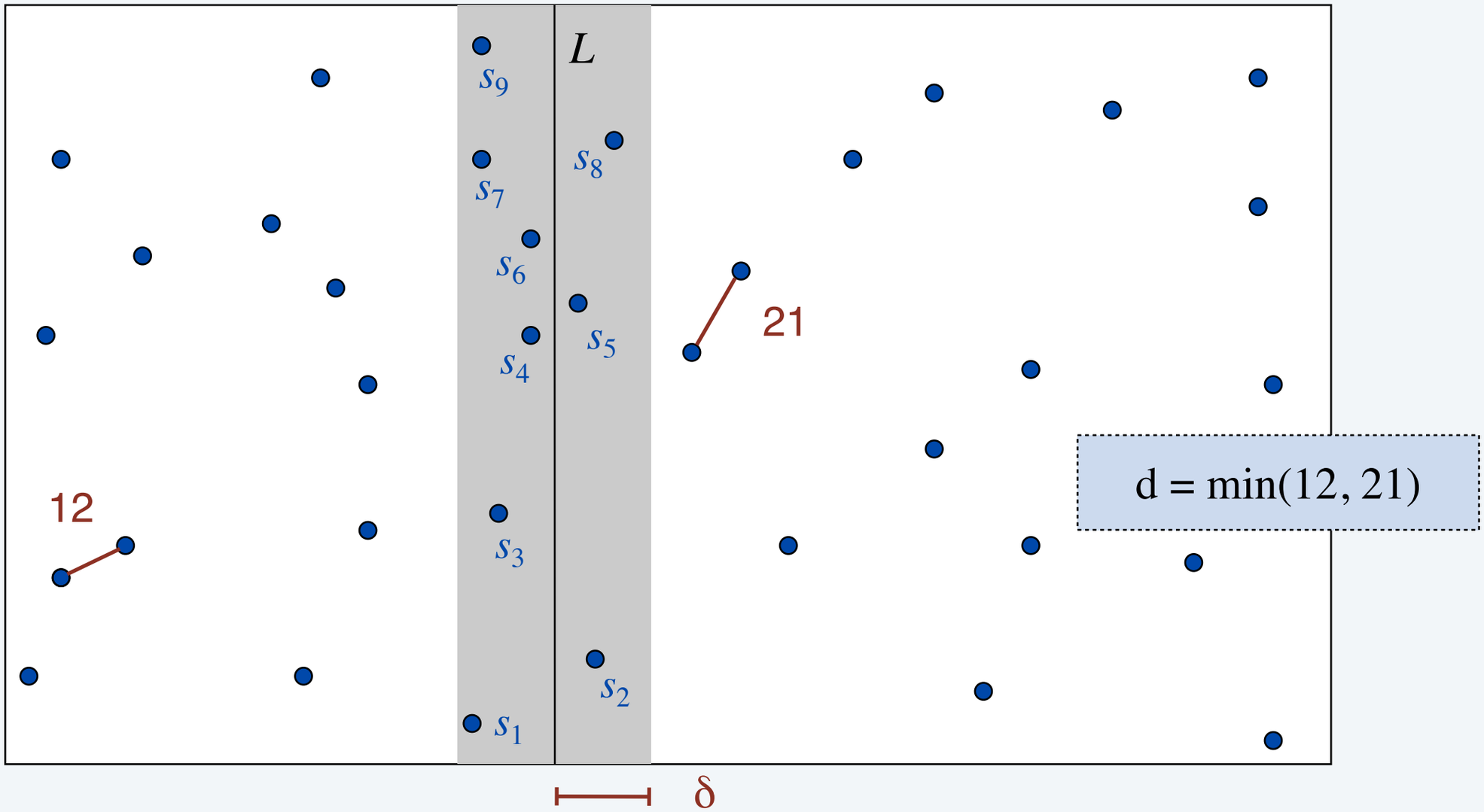
- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.



How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.
- Check distances of only those points within 7 positions in sorted list!

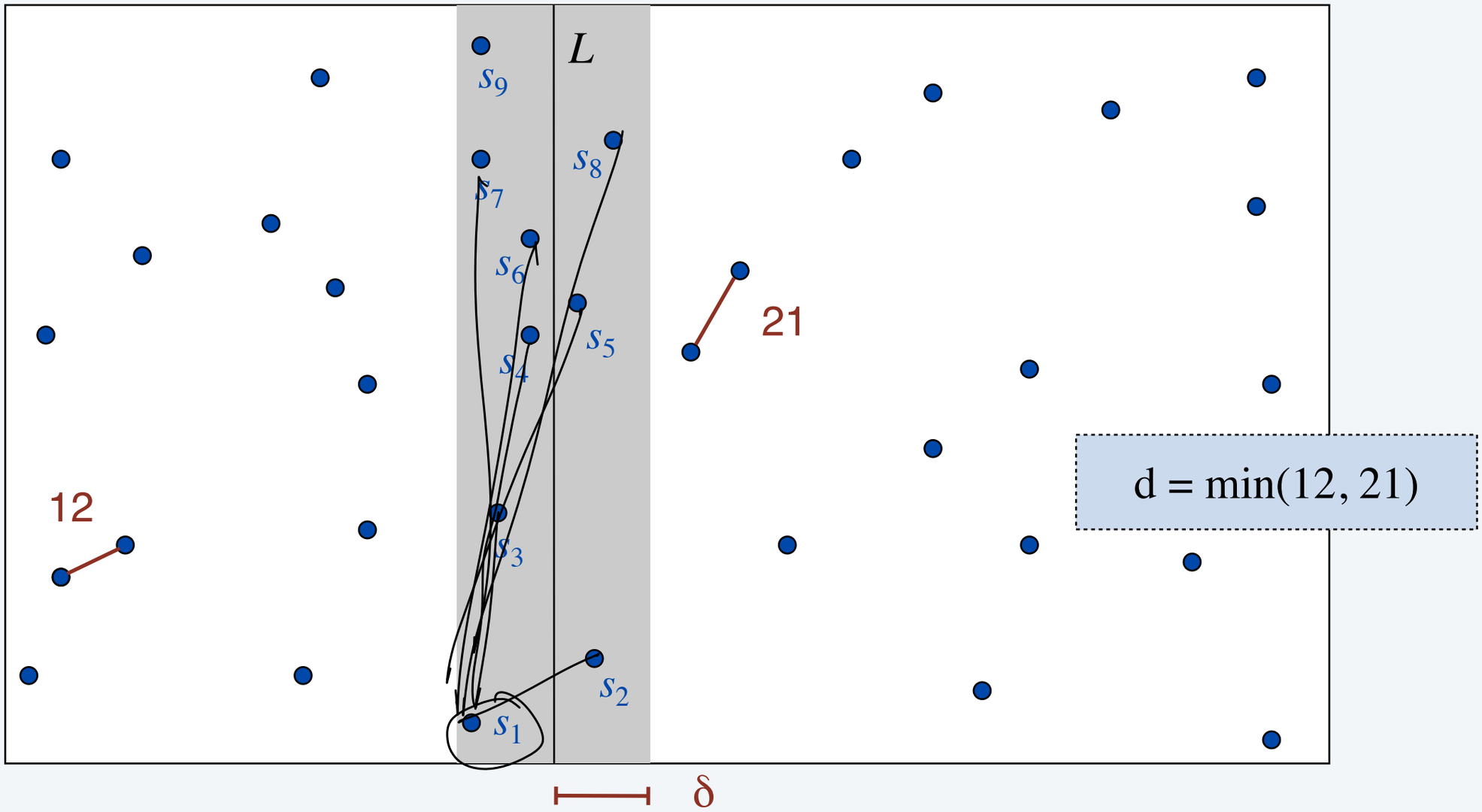


How to find closest pair with one point in each side?

Find closest pair with one point in each side, assuming that distance < d.

- Observation: suffices to consider only those points within d of line L.
- Sort points in $2\,\mathrm{d}$-strip by their y-coordinate.
- Check distances of only those points within 7 positions in sorted list!

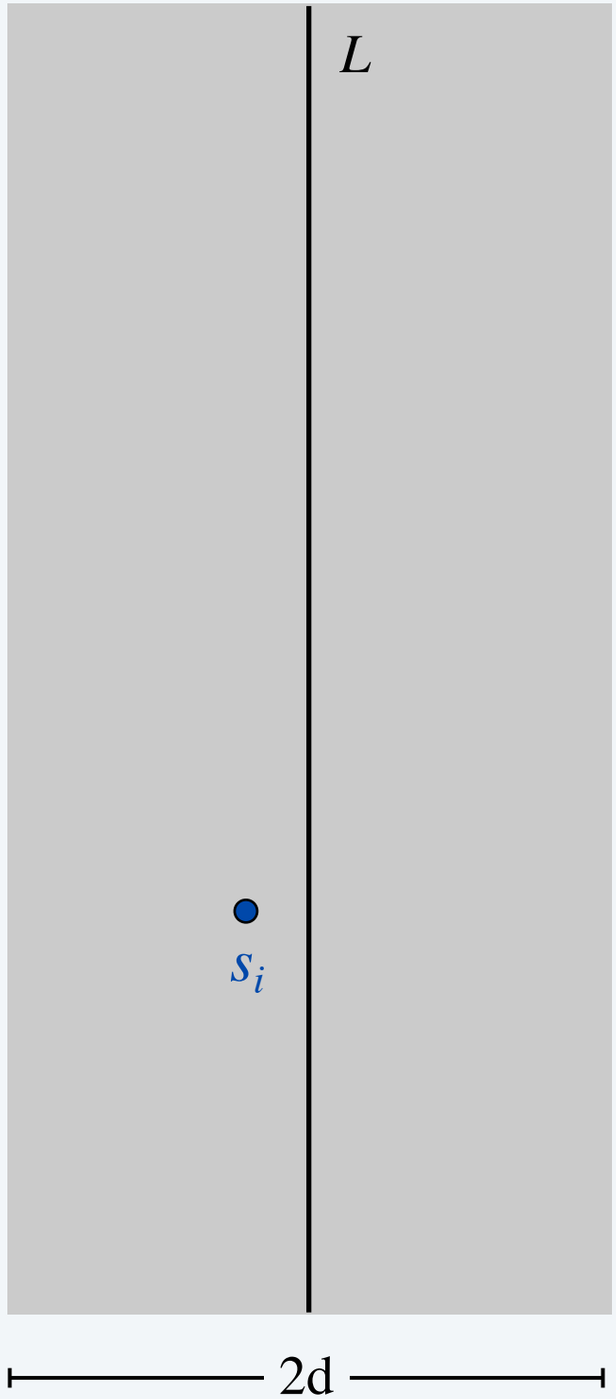
why?



How to find closest pair with one point in each side?

How to find closest pair with one point in each side?

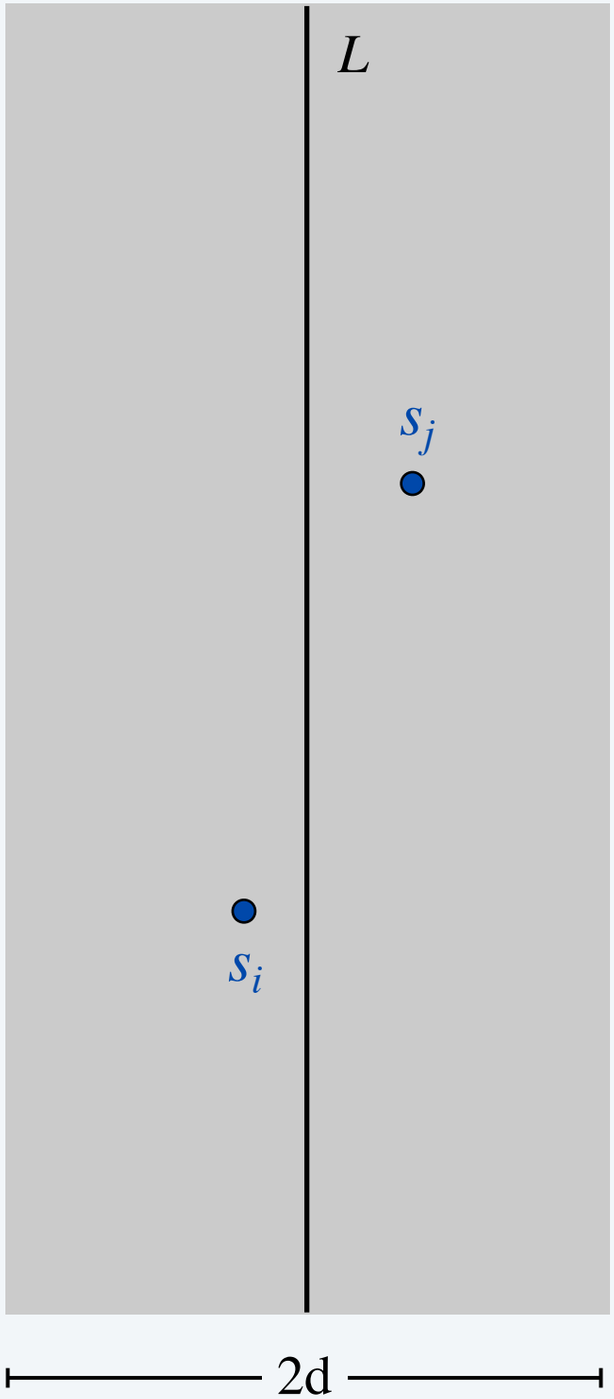
Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.



How to find closest pair with one point in each side?

Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

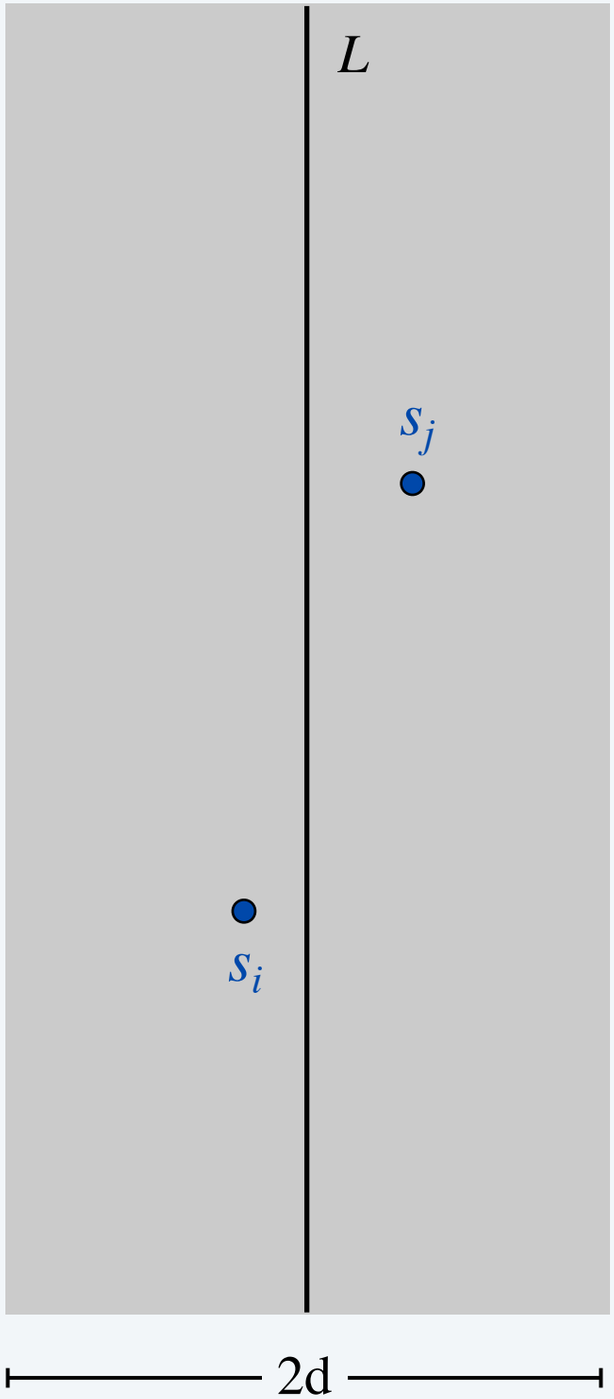


How to find closest pair with one point in each side?

Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.



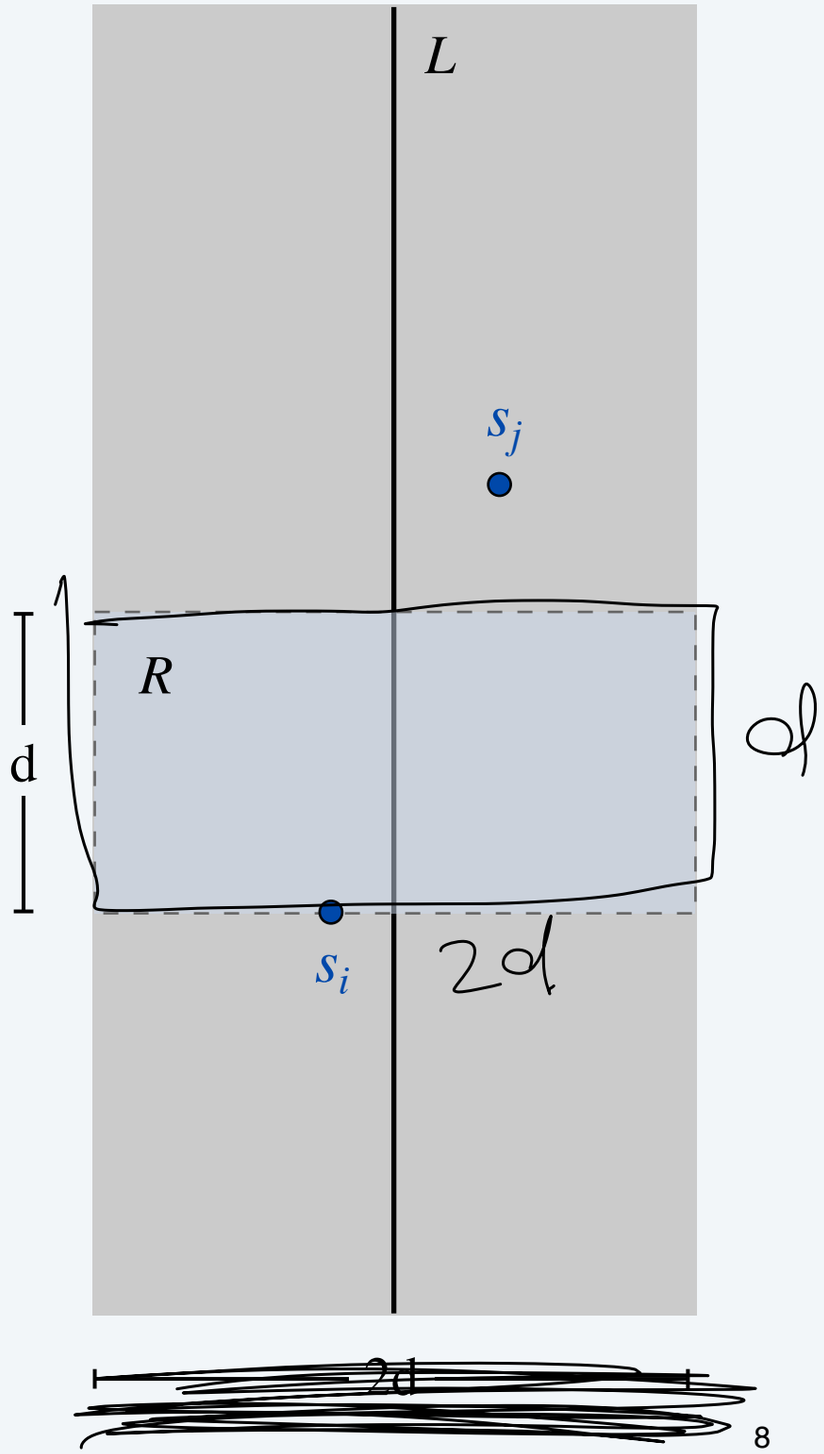
How to find closest pair with one point in each side?

Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.

- Consider the $2\mathrm{d}$-by-d rectangle R in strip whose min y-coordinate is y-coordinate of s_i.



How to find closest pair with one point in each side?

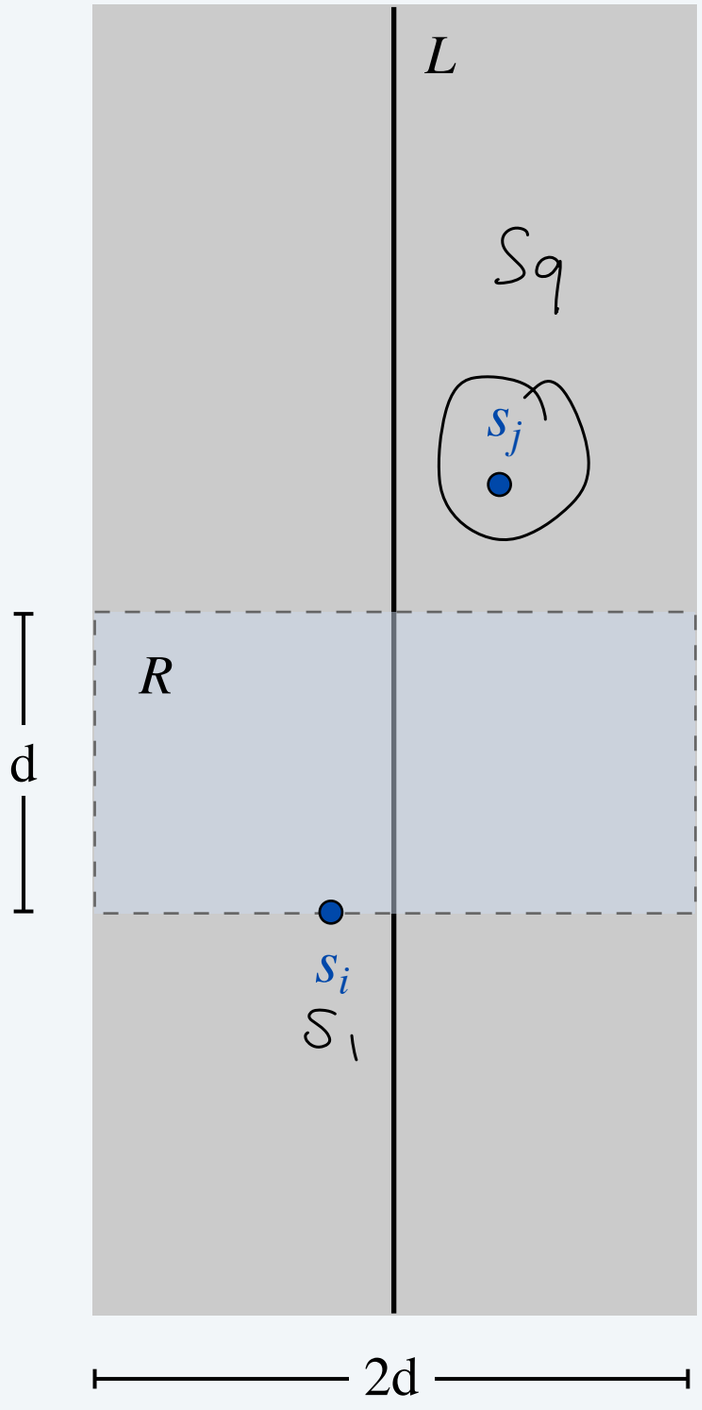
Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.

- Consider the $2\mathrm{d}$-by-d rectangle R in strip whose min y-coordinate is y-coordinate of s_i.
- Distance between s_i and any point s_j above R is $\geq \mathrm{d}$.

claim



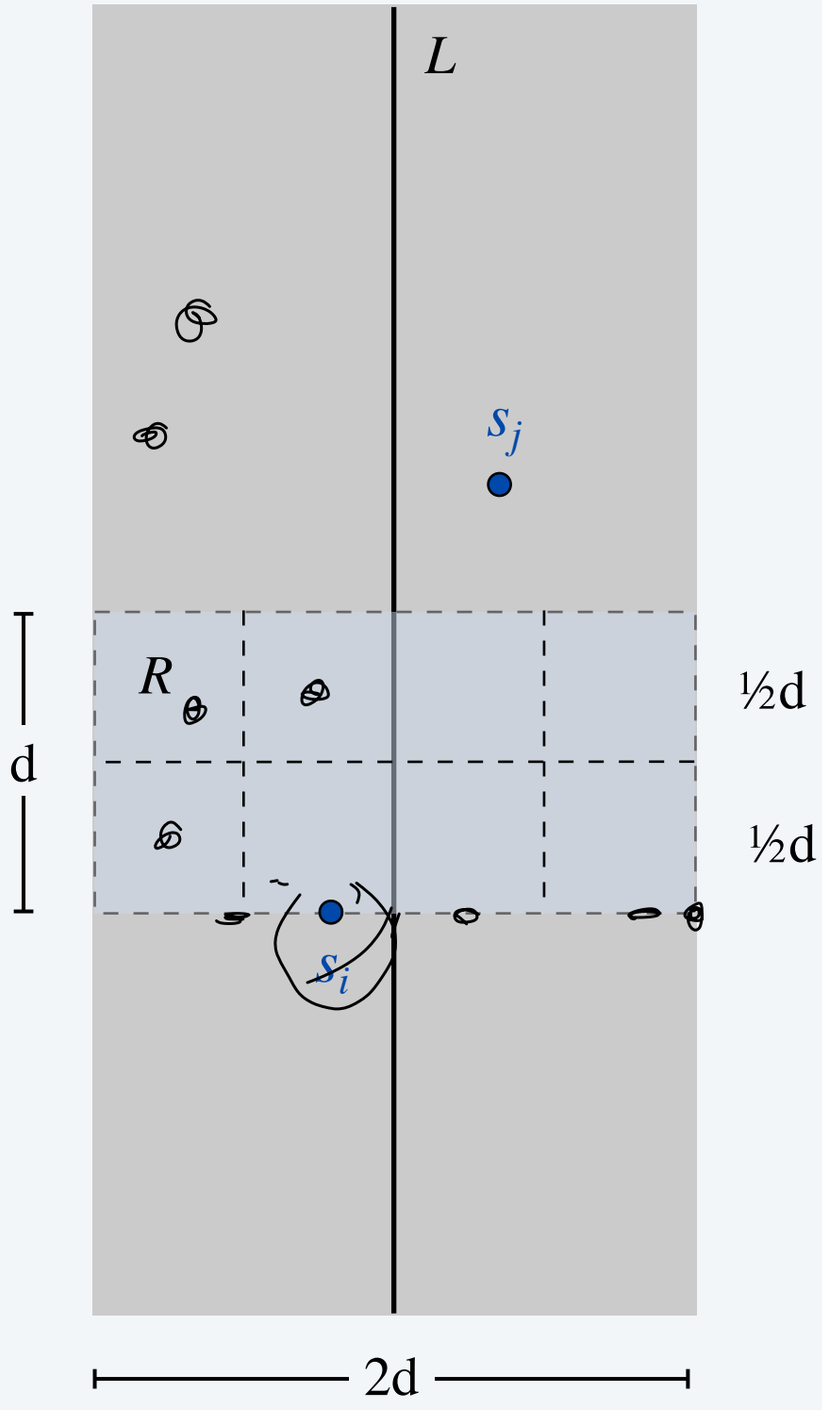
How to find closest pair with one point in each side?

Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.

- Consider the $2\mathrm{d}$-by-d rectangle R in strip whose min y-coordinate is y-coordinate of s_i.
- Distance between s_i and any point s_j above R is $\geq \mathrm{d}$.
- Subdivide R into 8 squares.



How to find closest pair with one point in each side?

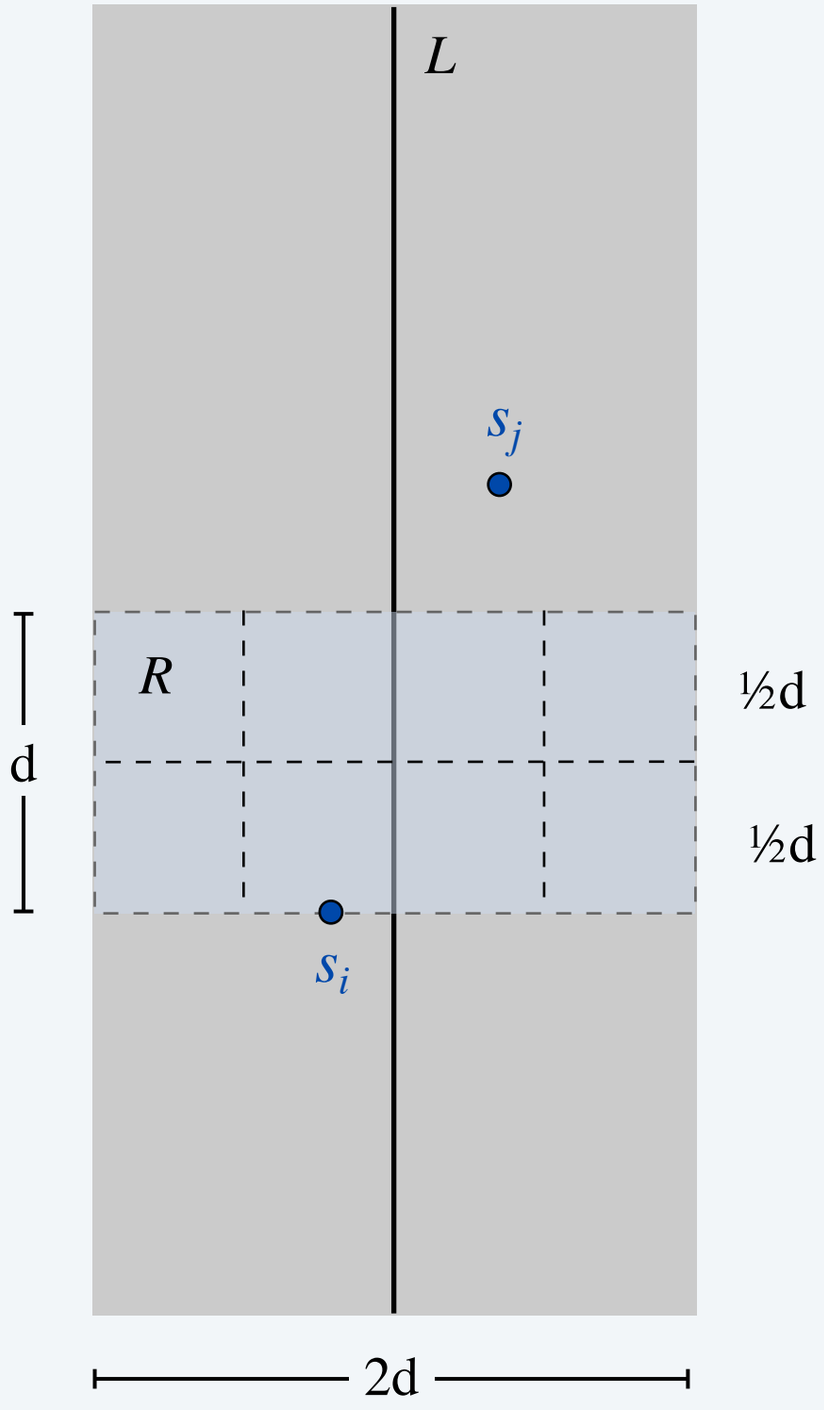
Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.

- Consider the $2\mathrm{d}$-by-d rectangle R in strip whose min y-coordinate is y-coordinate of s_i.
- Distance between s_i and any point s_j above R is $\geq \mathrm{d}$.
- Subdivide R into 8 squares.
- At most 1 point per square.

diameter of square is $d/\sqrt{2} < d$



How to find closest pair with one point in each side?

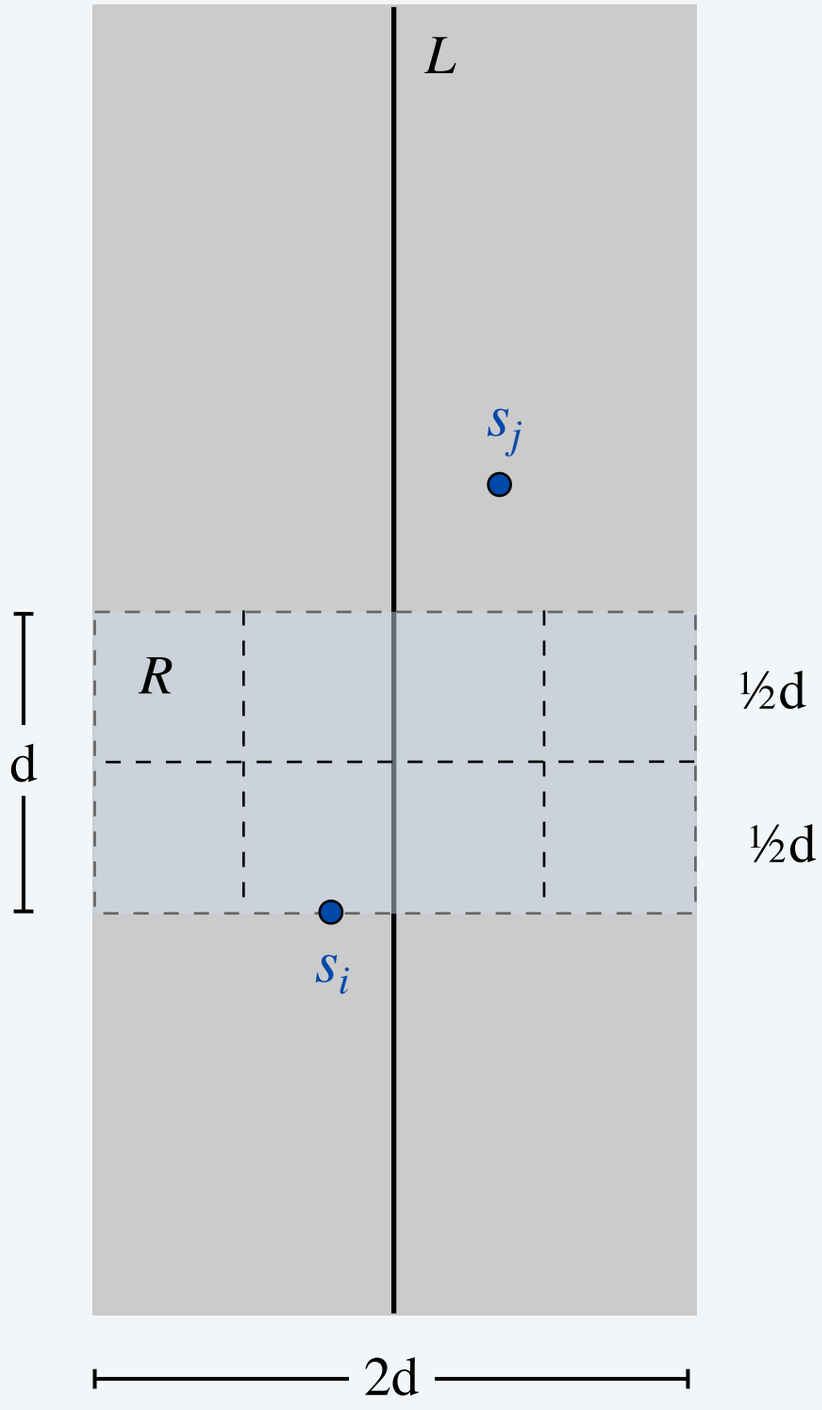
Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.

- Consider the $2\mathrm{d}$-by-d rectangle R in strip whose min y-coordinate is y-coordinate of s_i.
- Distance between s_i and any point s_j above R is $\geq \mathrm{d}$.
- Subdivide R into 8 squares.
- At most 1 point per square.
- At most 7 points other than s_i can be in R. ▪

diameter of square is $d/\sqrt{2} < d$



How to find closest pair with one point in each side?

Def. Let s_i be the point in the $2\,\mathrm{d}$-strip with the i^{th} smallest y-coordinate.

Claim. If $|j - i| > 7$, then the distance between s_i and s_j is at least d.

Pf.

- Consider the $2\mathrm{d}$-by-d rectangle R in strip whose min y-coordinate is y-coordinate of s_i.
- Distance between s_i and any point s_j above R is $\geq \mathrm{d}$.
- Subdivide R into 8 squares.
- At most 1 point per square.
- At most 7 points other than s_i can be in R. ▪

diameter of square is $d/\sqrt{2} < d$

constant can be improved with more refined geometric packing argument