

<http://algs4.cs.princeton.edu>

2D TREE DEMO

- ▶ *insertion*
- ▶ *range search*
- ▶ *nearest neighbor*



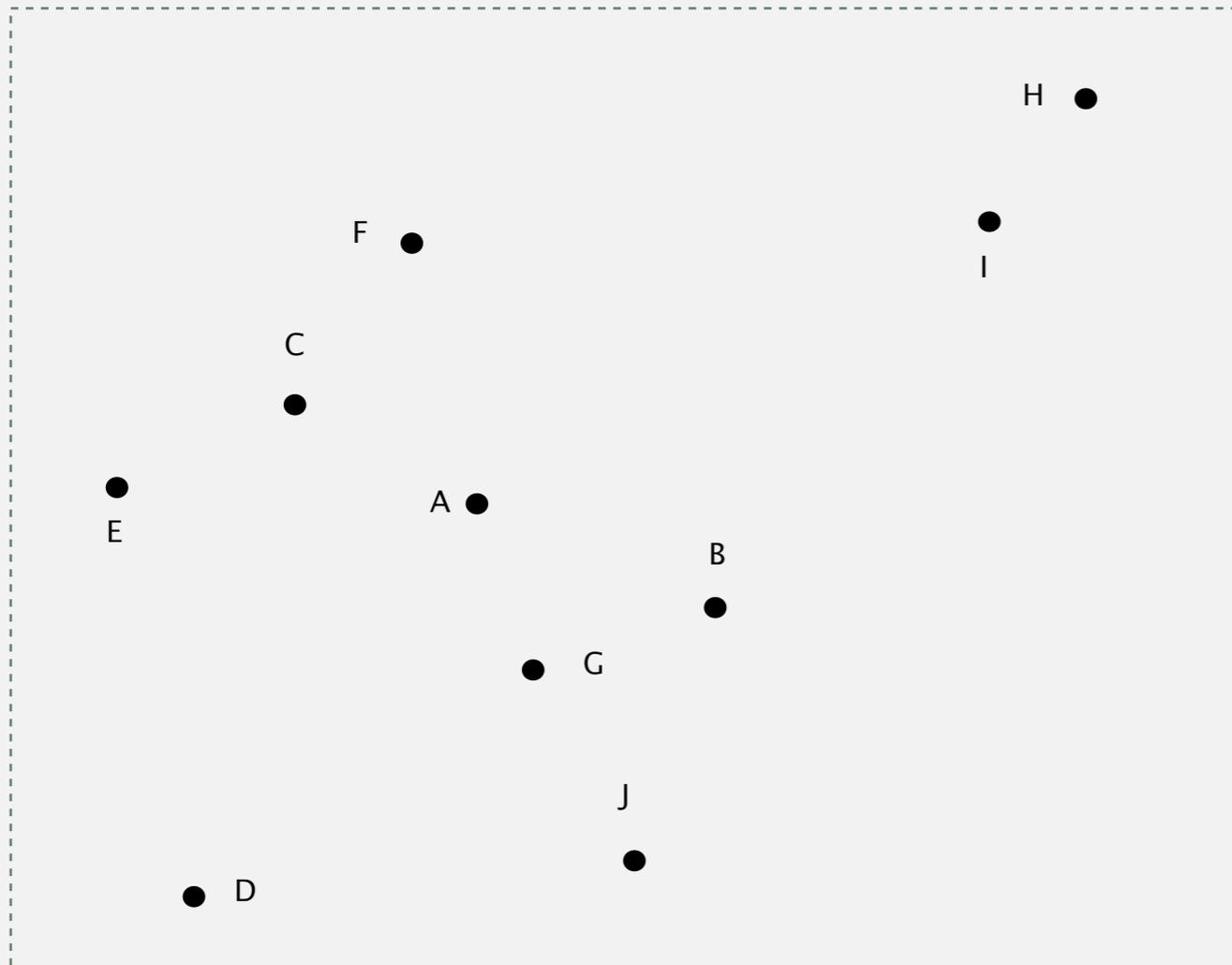
<http://algs4.cs.princeton.edu>

2D TREE DEMO

- ▶ *insertion*
- ▶ *range search*
- ▶ *nearest neighbor*

2d tree demo

Points to insert.



% more input10.txt

0.372 0.497

0.564 0.413

0.226 0.577

0.144 0.179

0.083 0.510

0.320 0.708

0.417 0.362

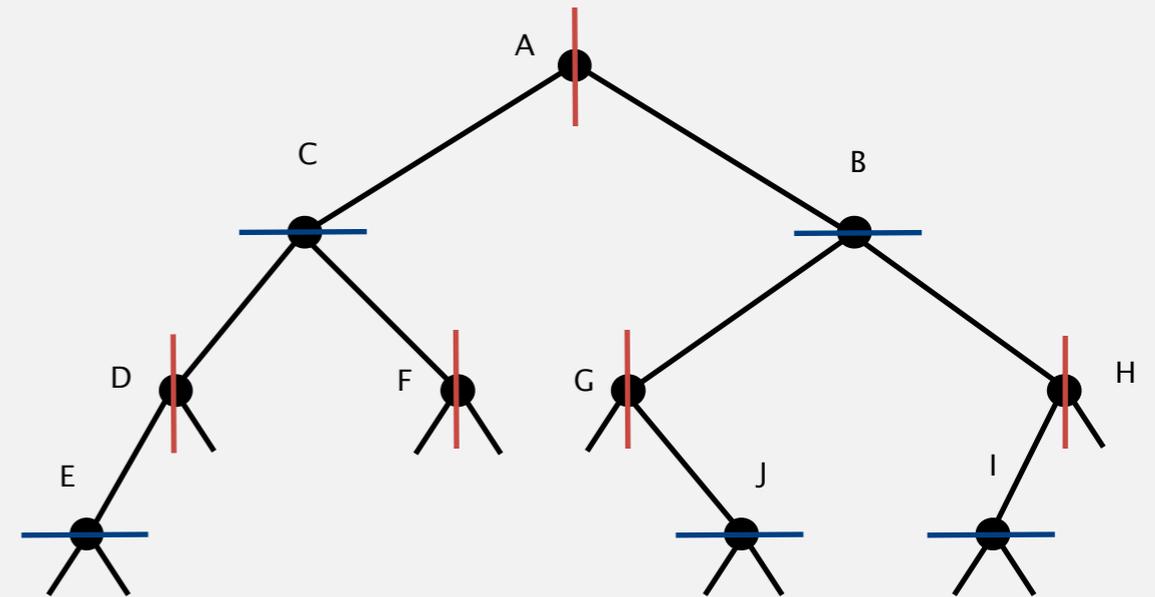
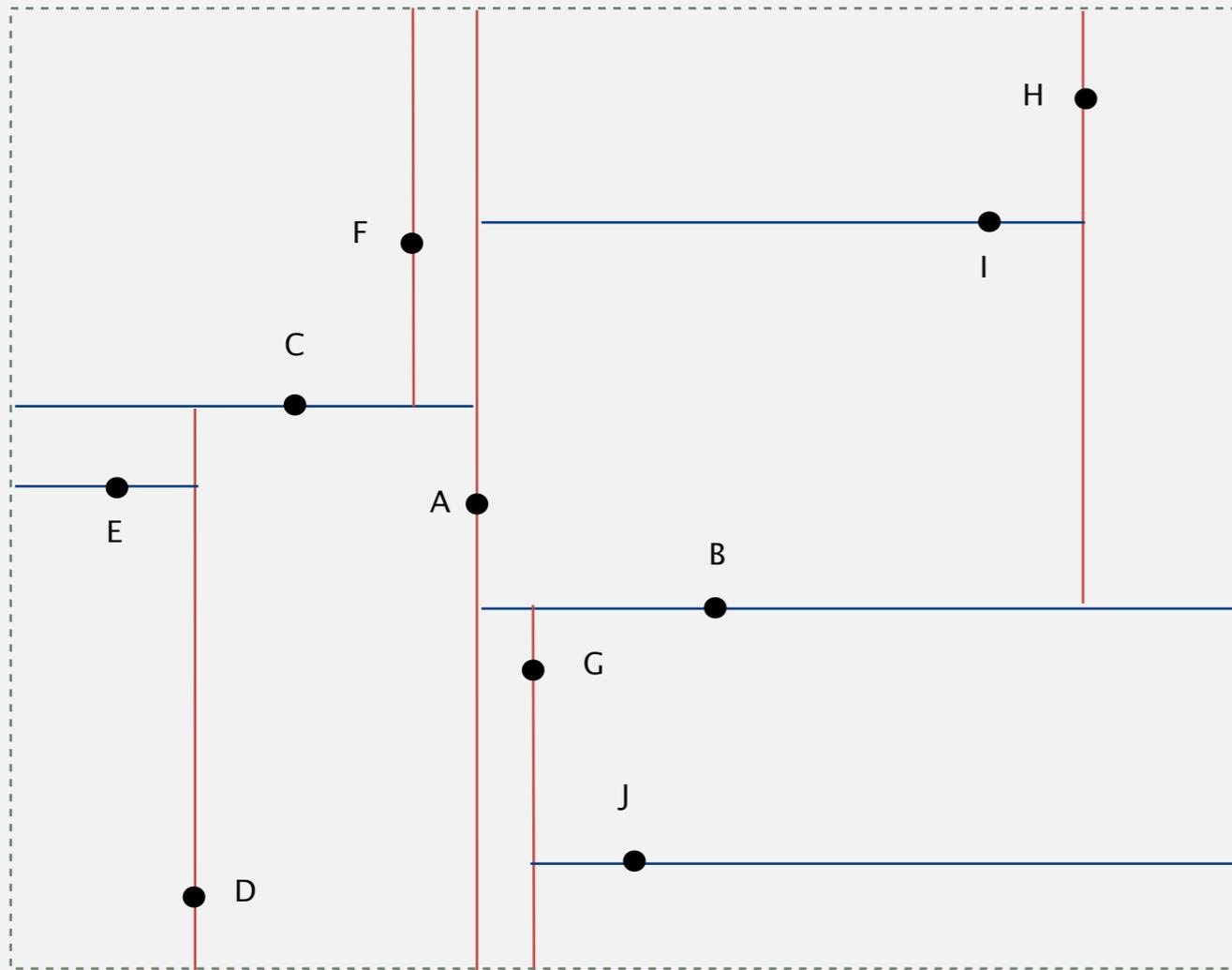
0.862 0.825

0.785 0.725

0.499 0.208

2d tree demo: insertion

Recursively partition plane into two halfplanes.



2D TREE DEMO

- ▶ *insertion*
- ▶ *range search*
- ▶ *nearest neighbor*



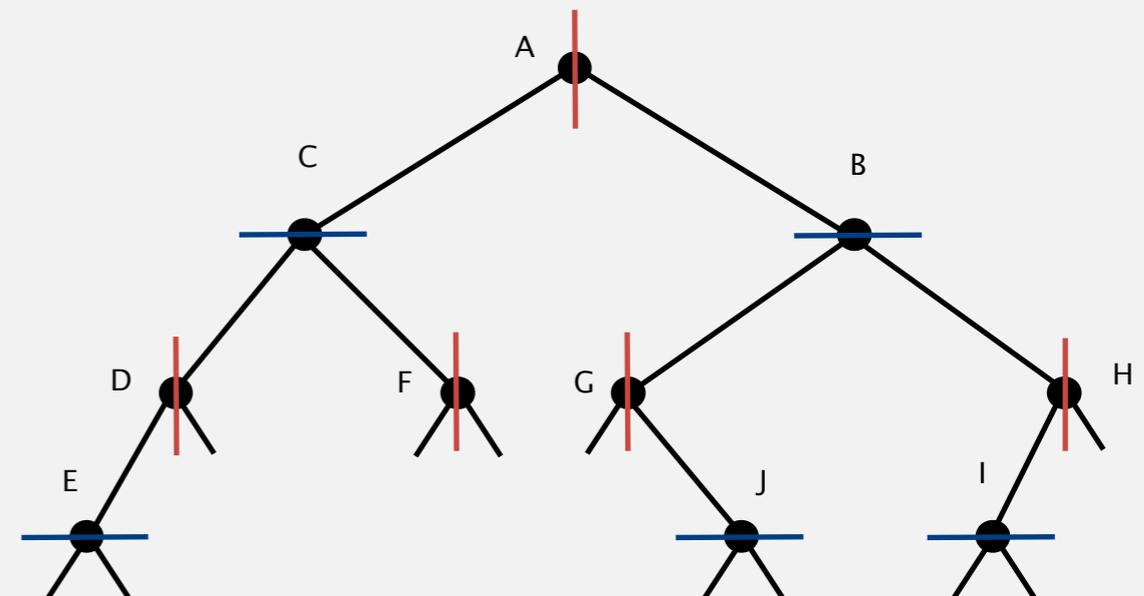
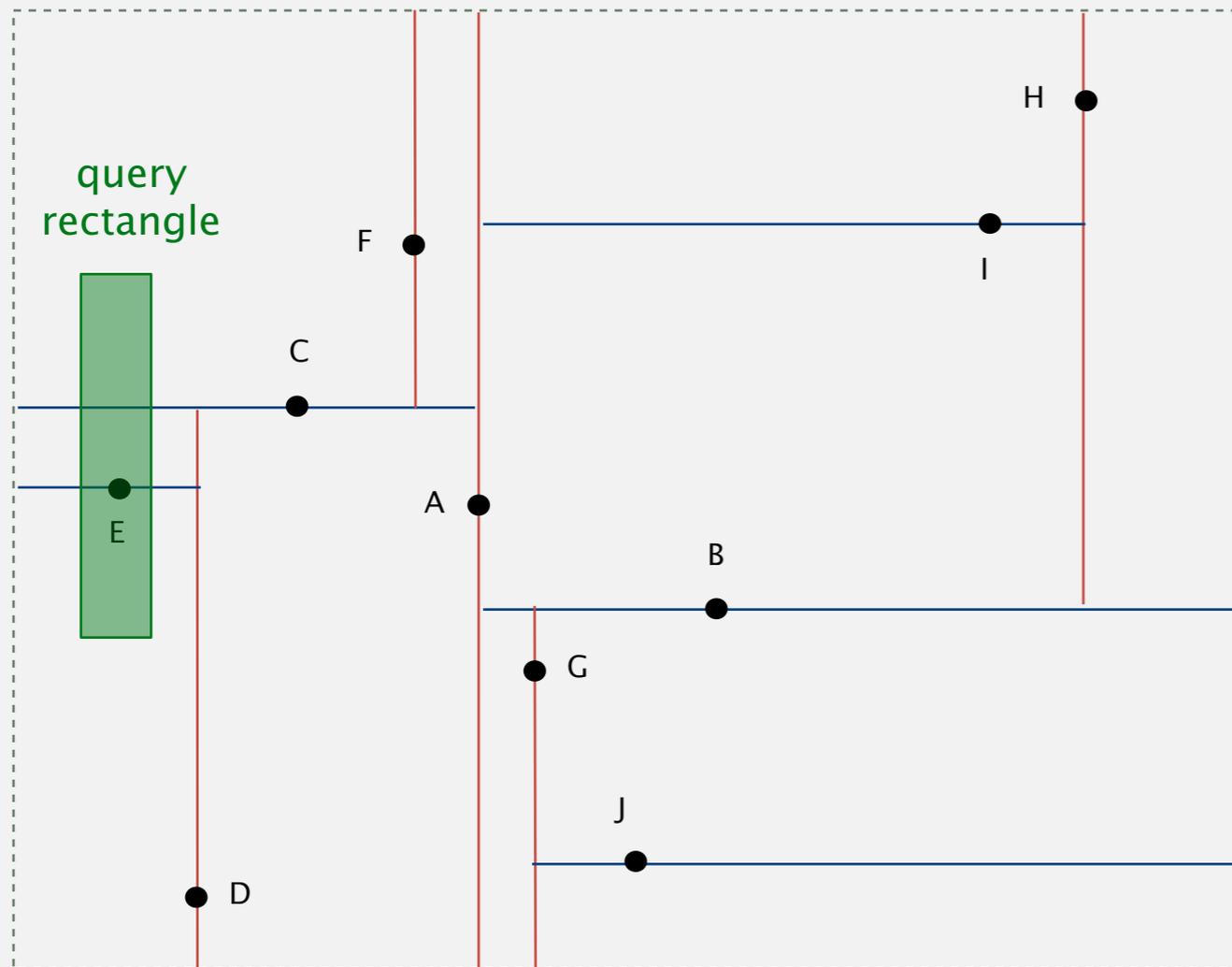
Algorithms

ROBERT SEDGEWICK | KEVIN WAYNE

<http://algs4.cs.princeton.edu>

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

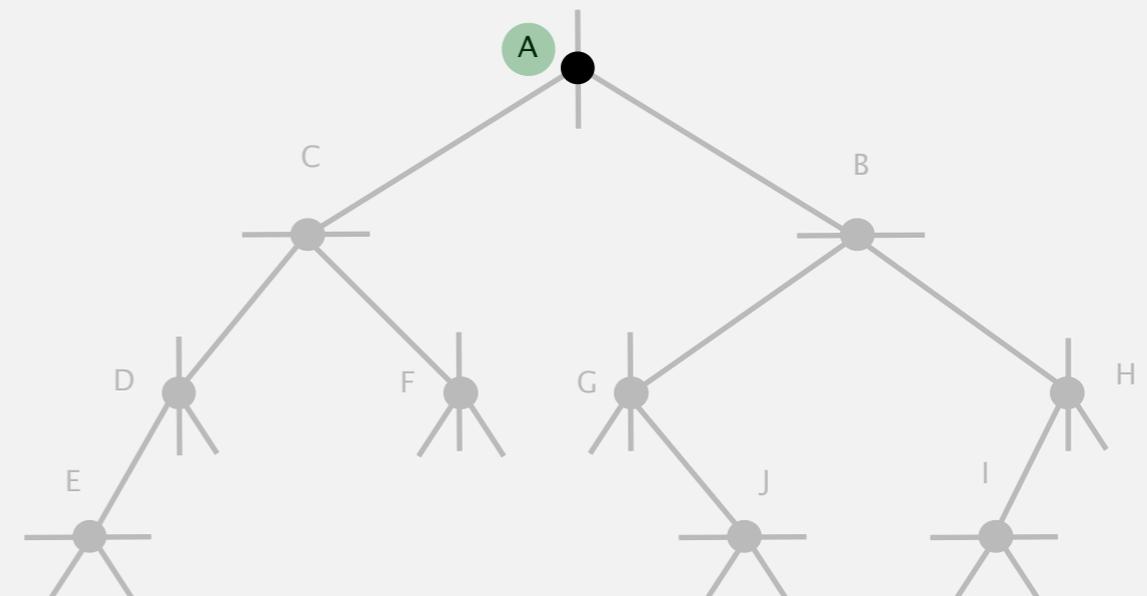
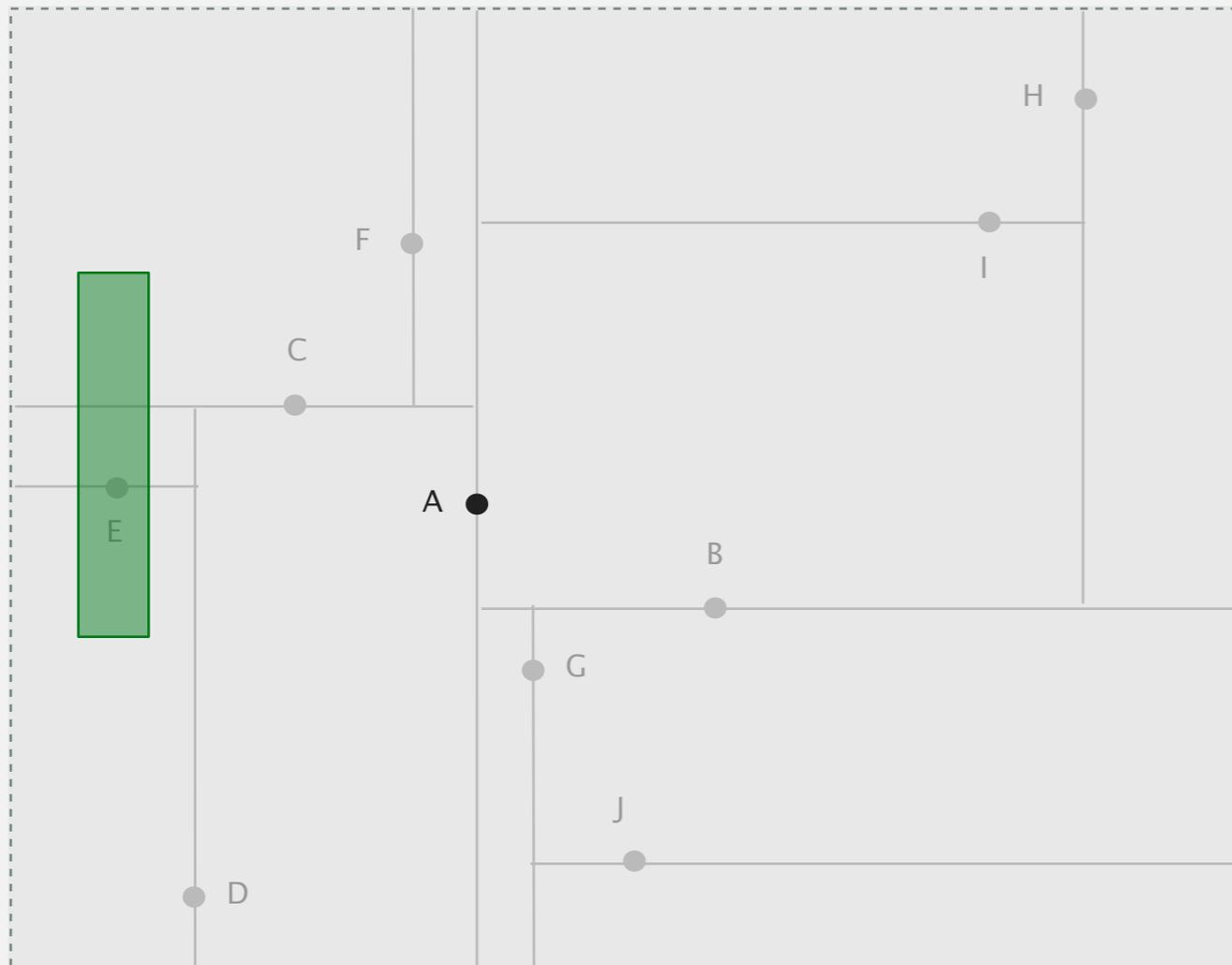


$(x_{\min}, y_{\min}, x_{\max}, y_{\max}) = (0.053, 0.393, 0.106, 0.683)$

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.

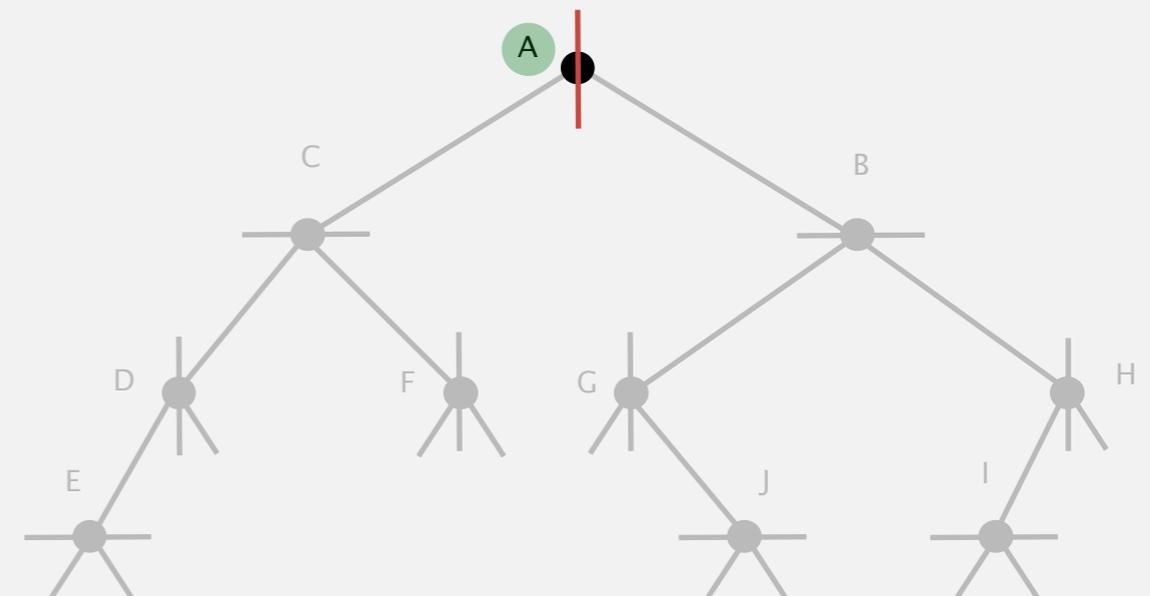
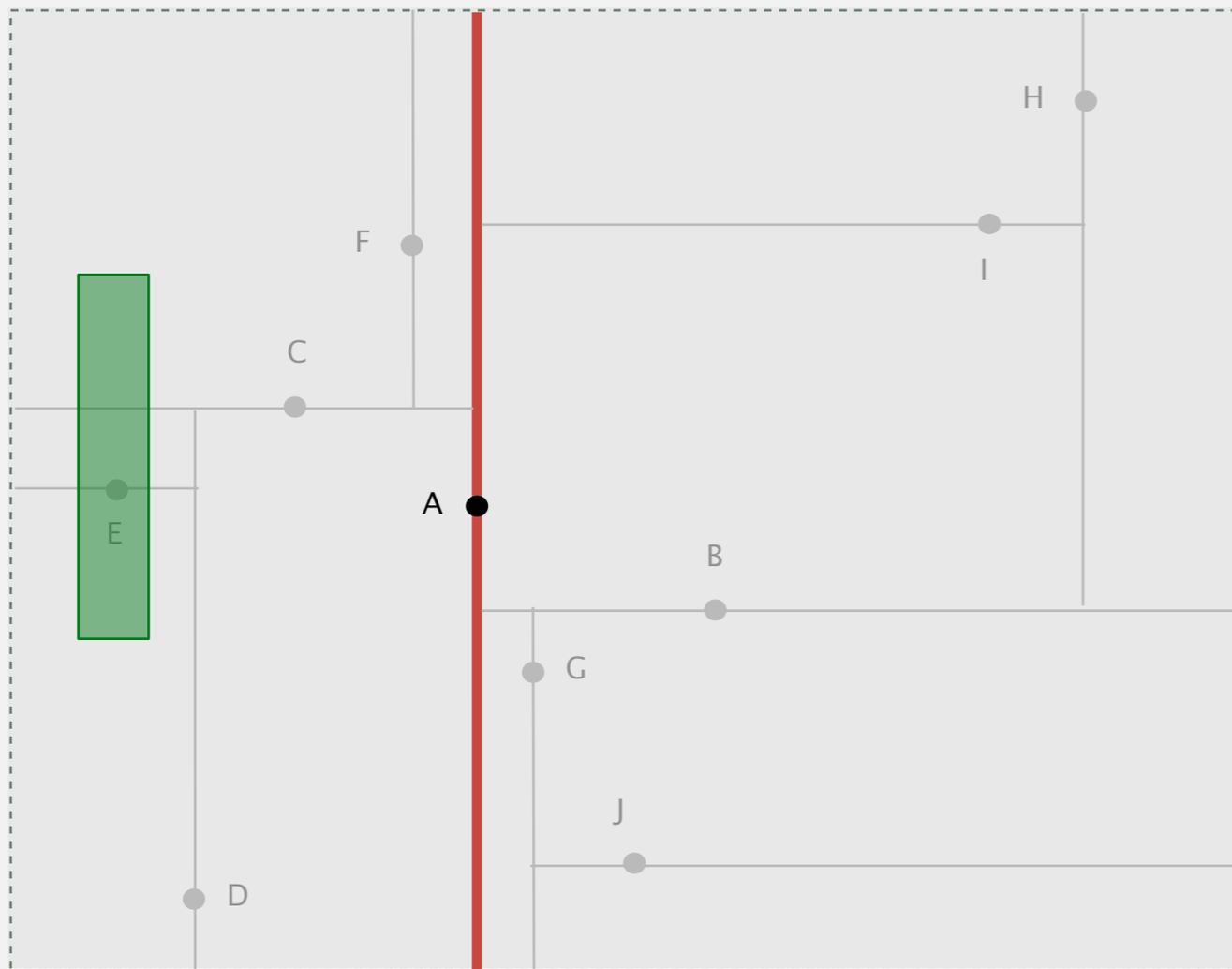


search root node
check if query rectangle contains point A

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

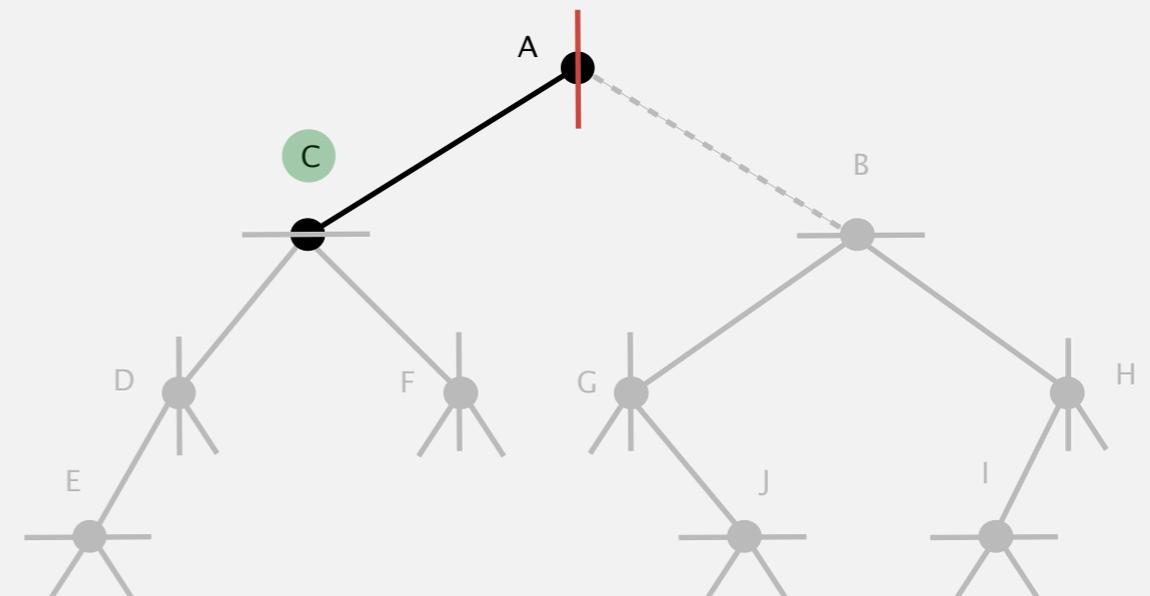
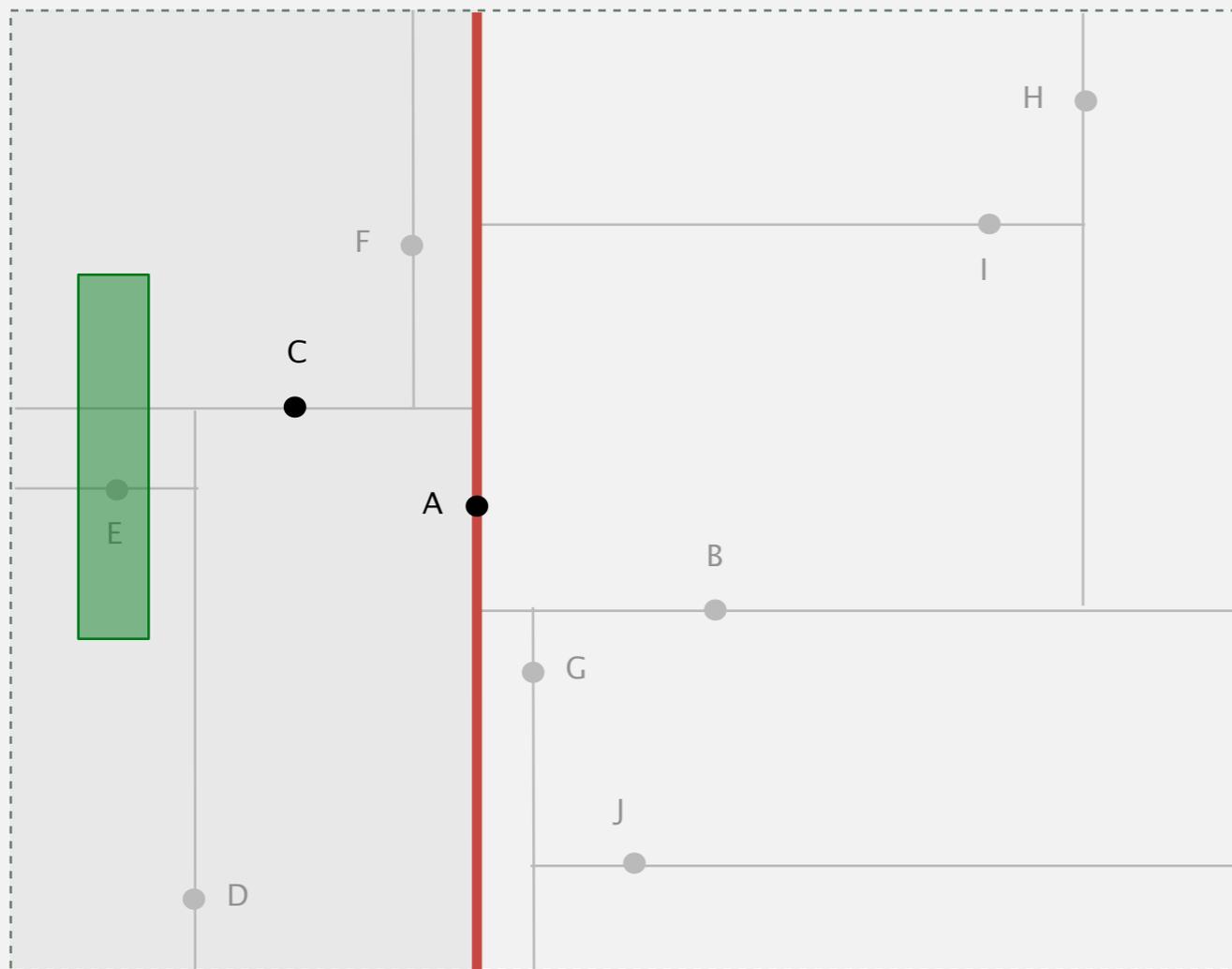


**query rectangle to left of splitting line
search only in left subtree**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

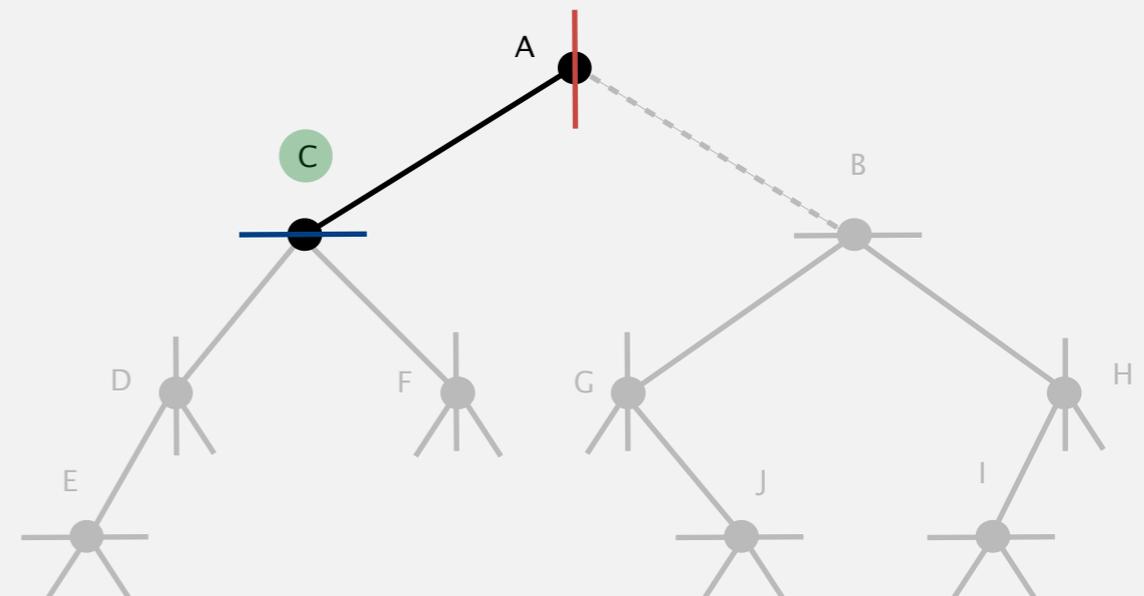
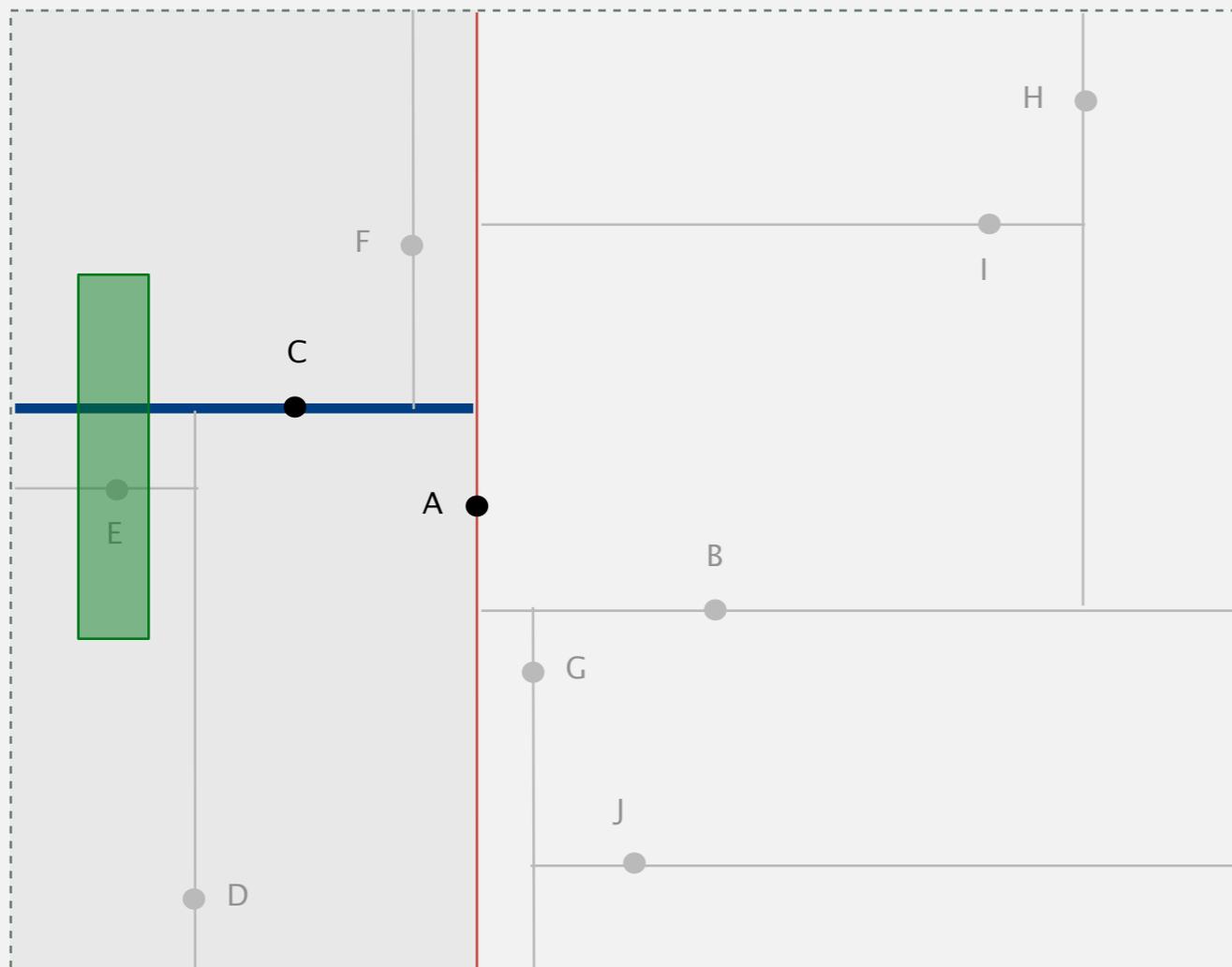


search left subtree
check if query rectangle contains point C

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

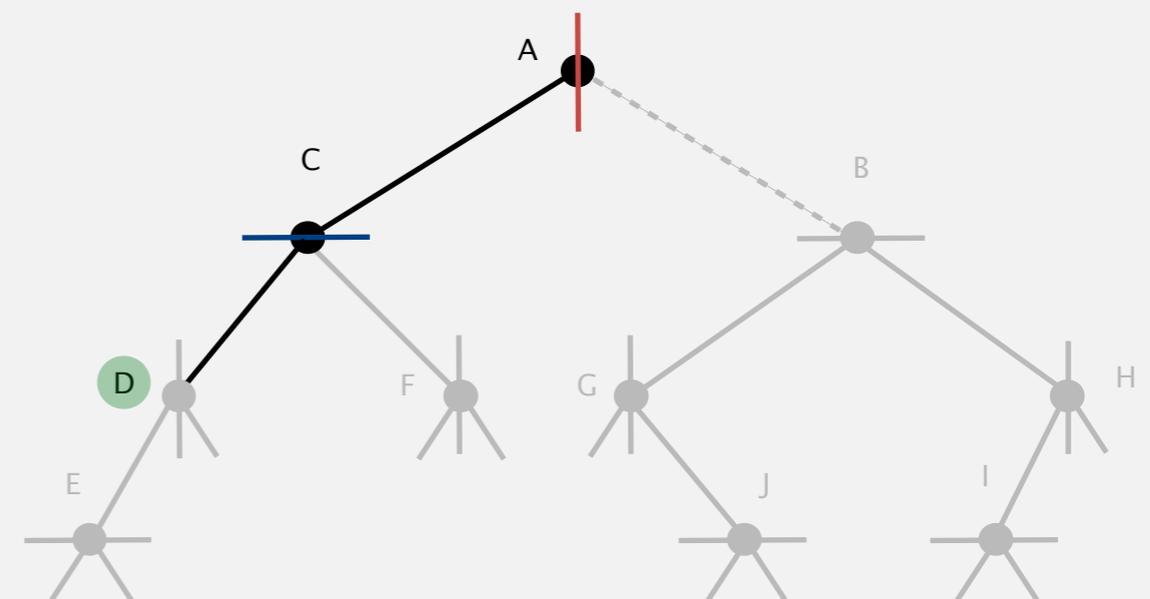
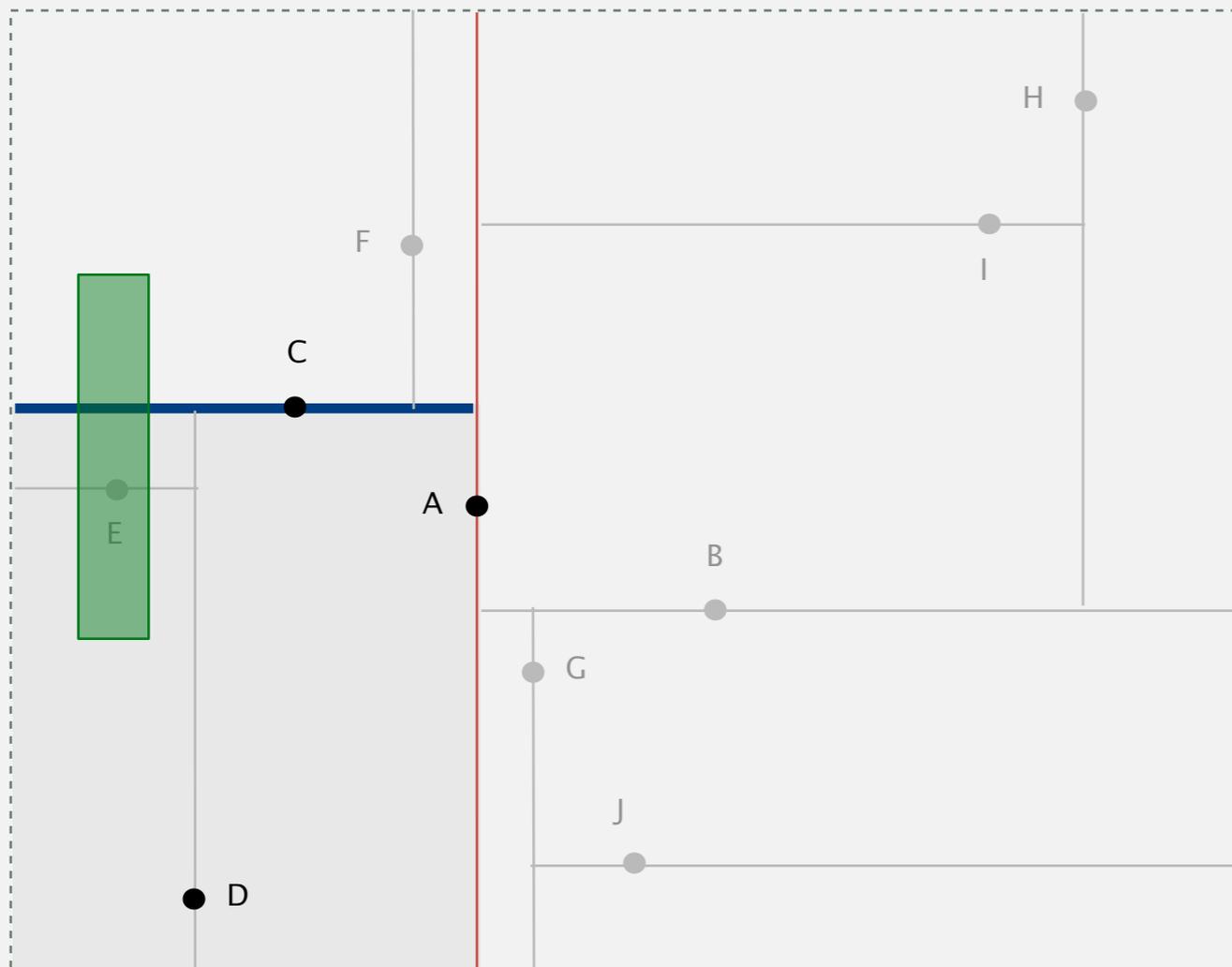


**query rectangle intersects splitting line
search bottom and top subtrees**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

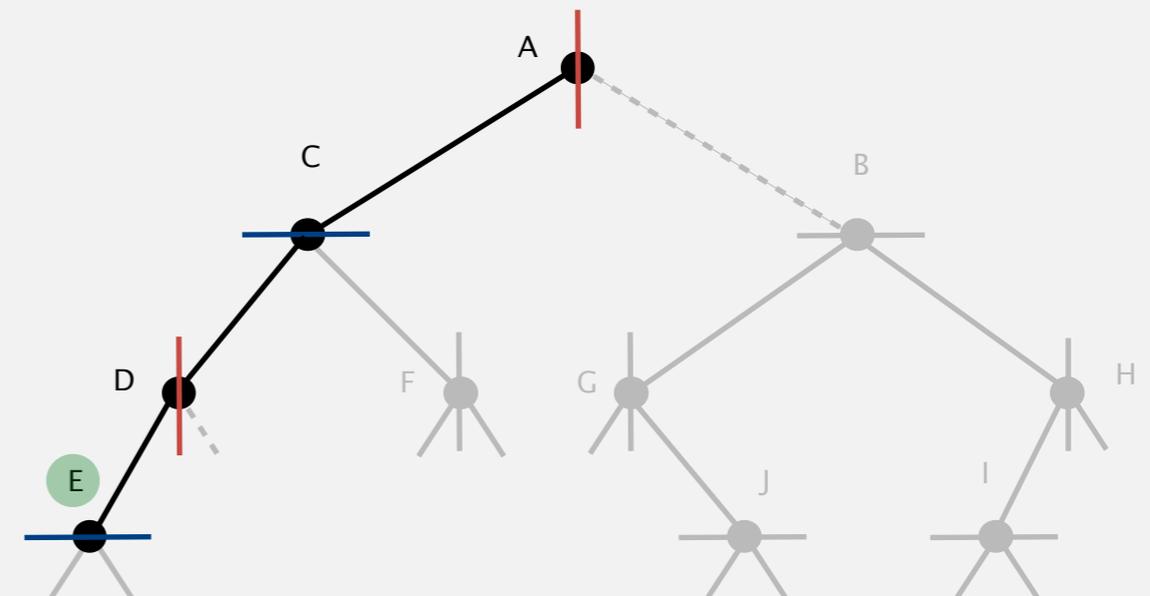
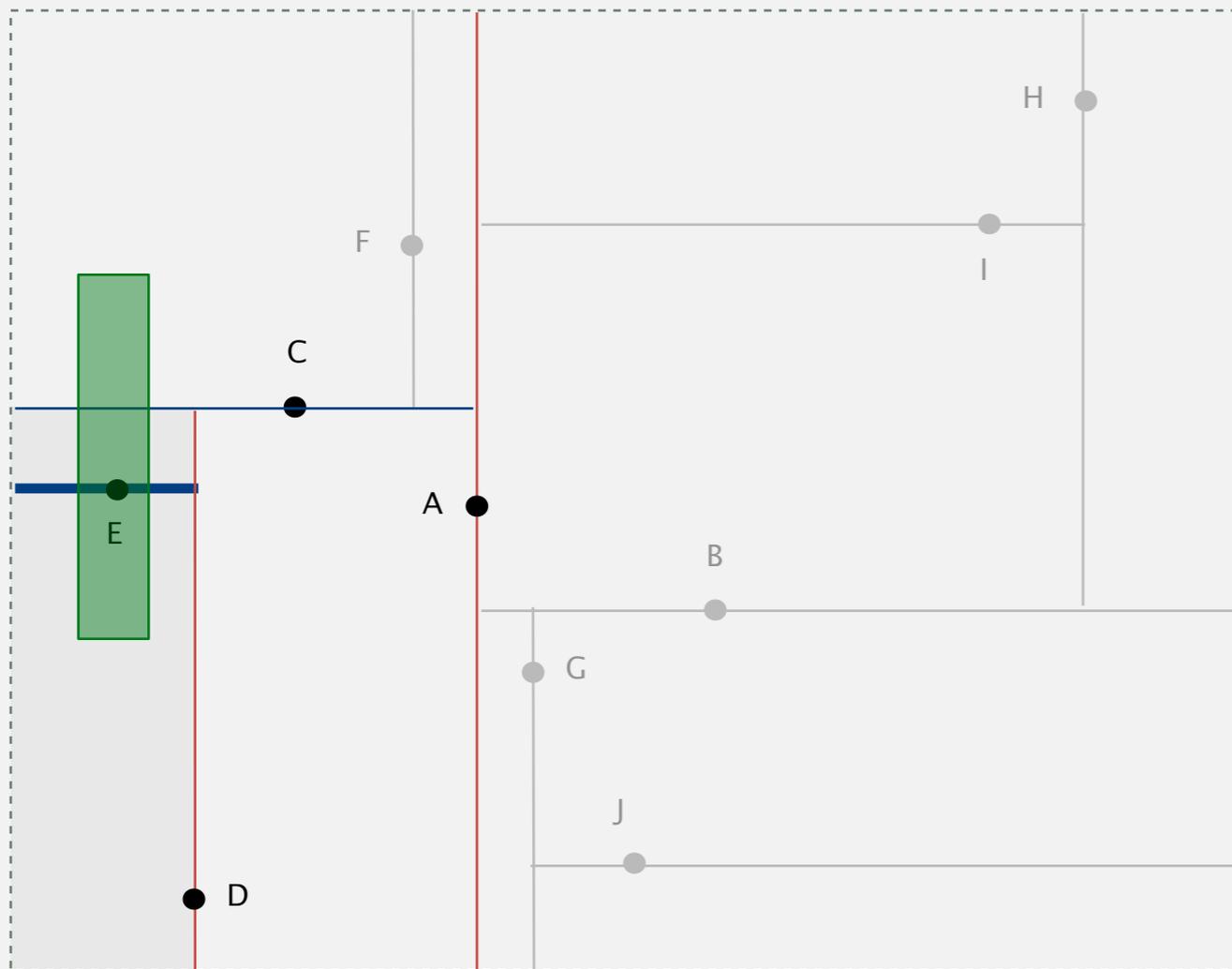


search left subtree
check if query rectangle contains point D

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

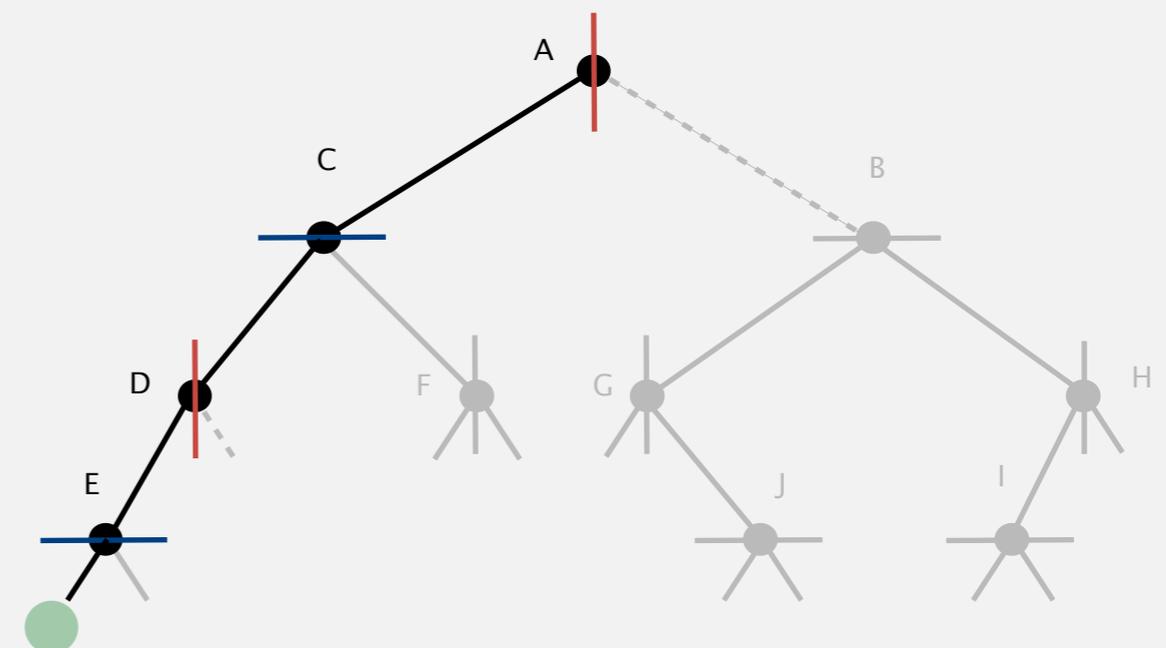
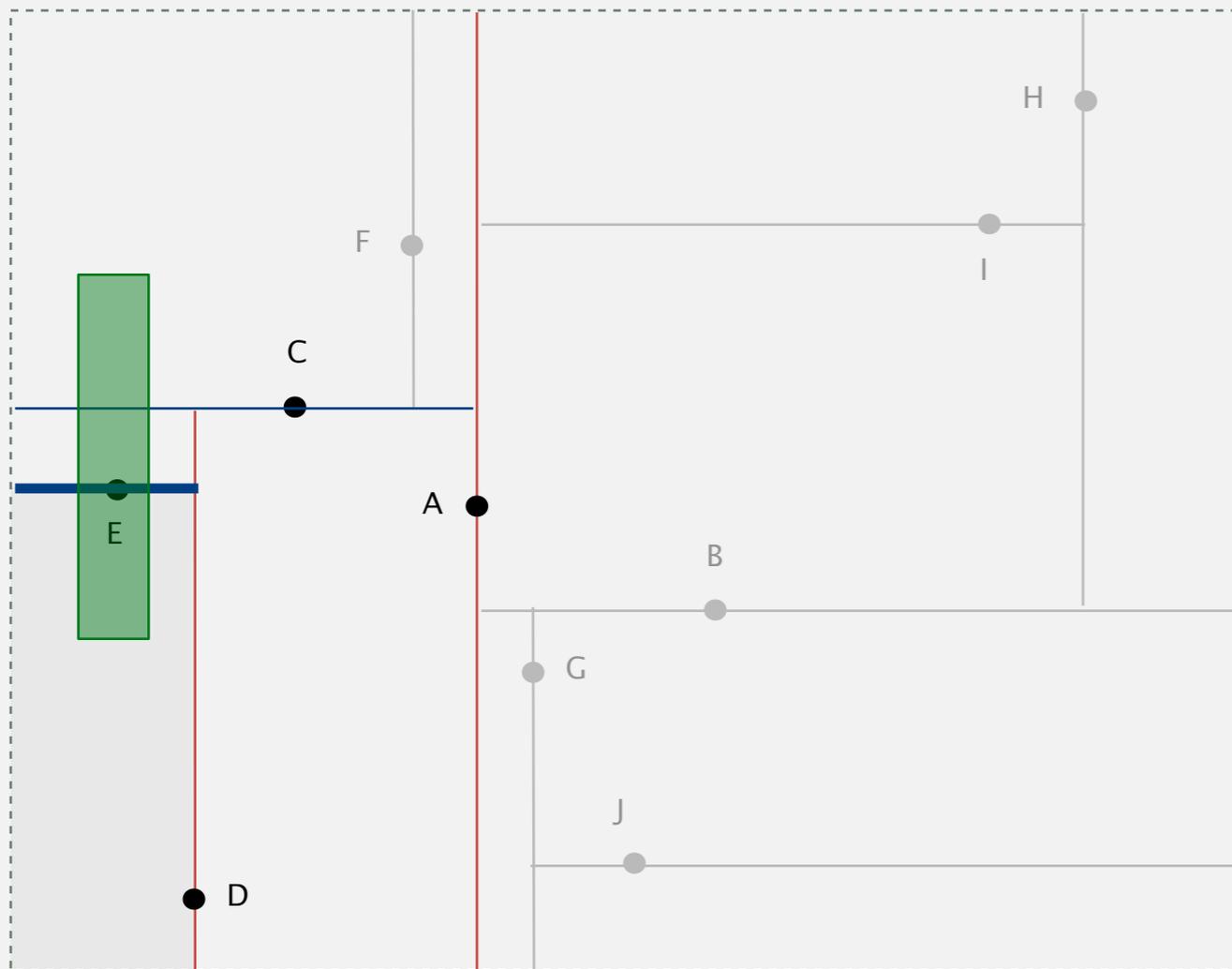


**query rectangle intersects splitting line
search bottom and top subtrees**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

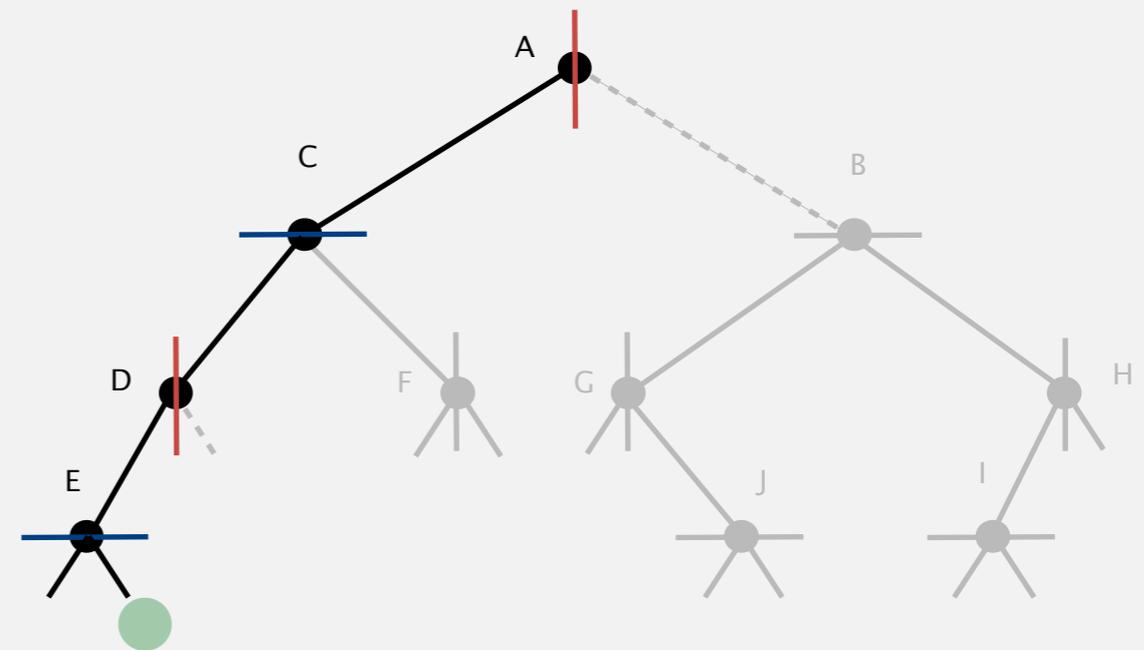
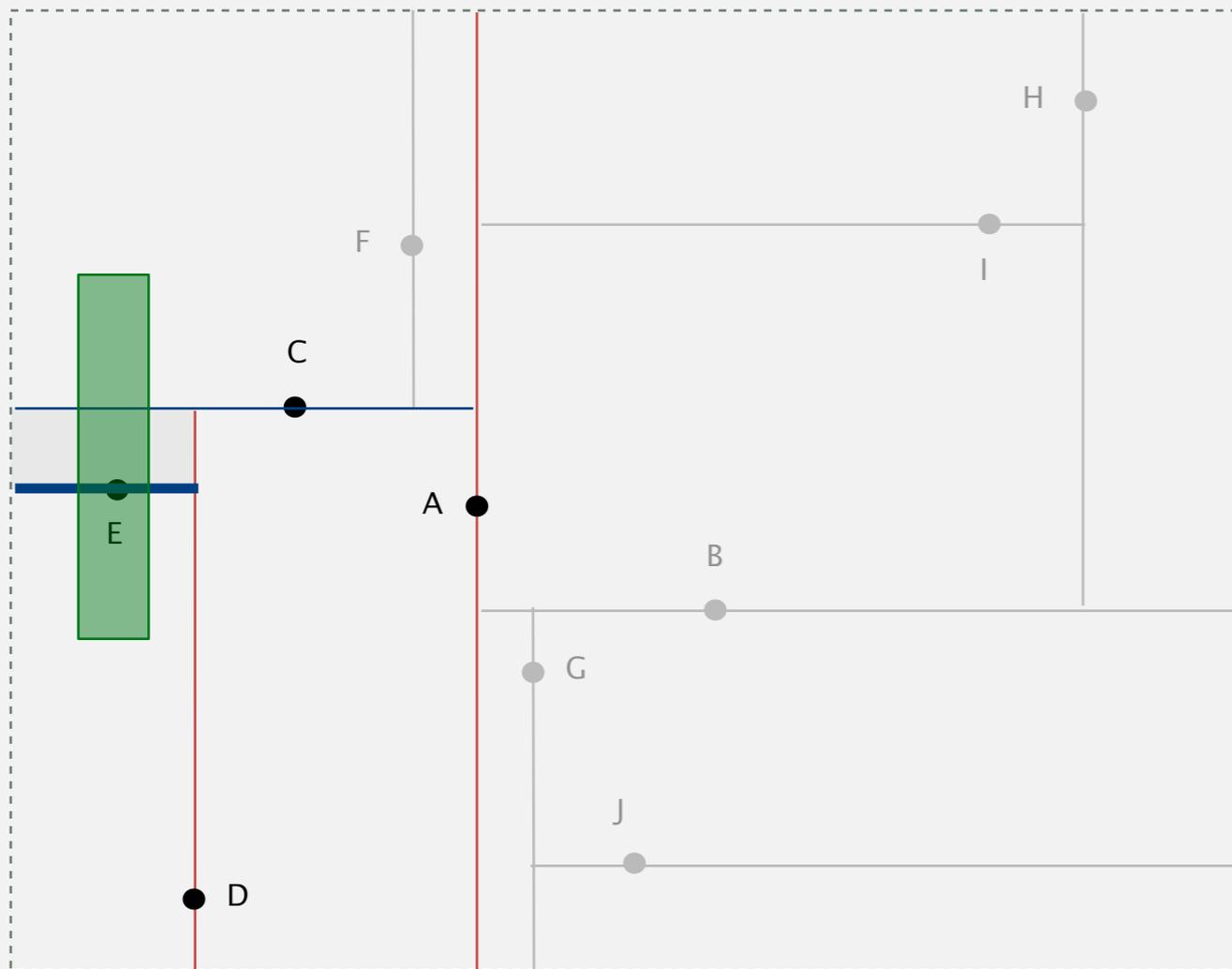


**search bottom subtree
stop since empty**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

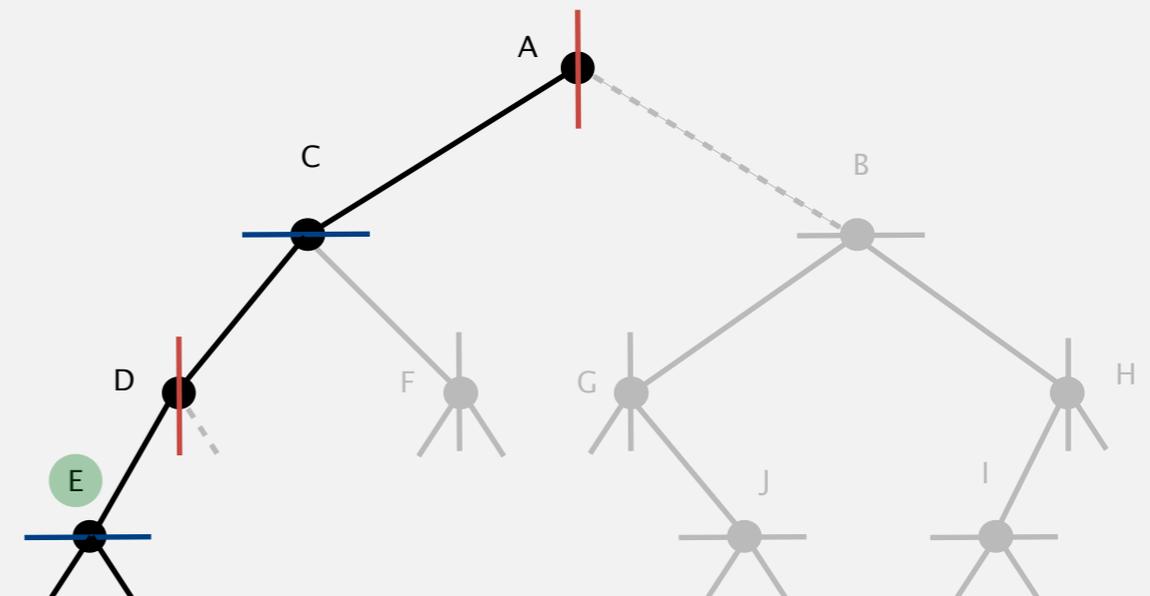
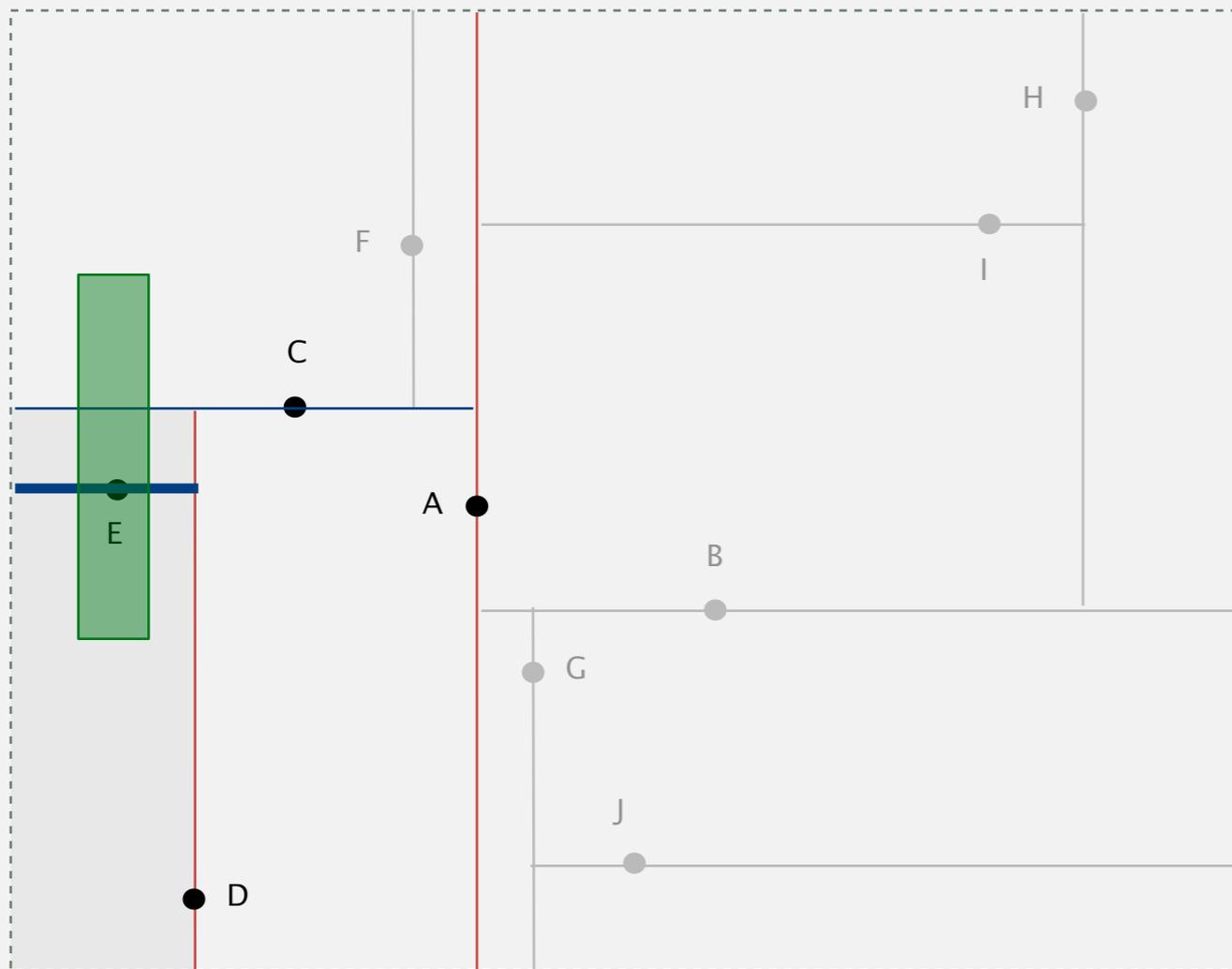


**search top subtree
stop since empty**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

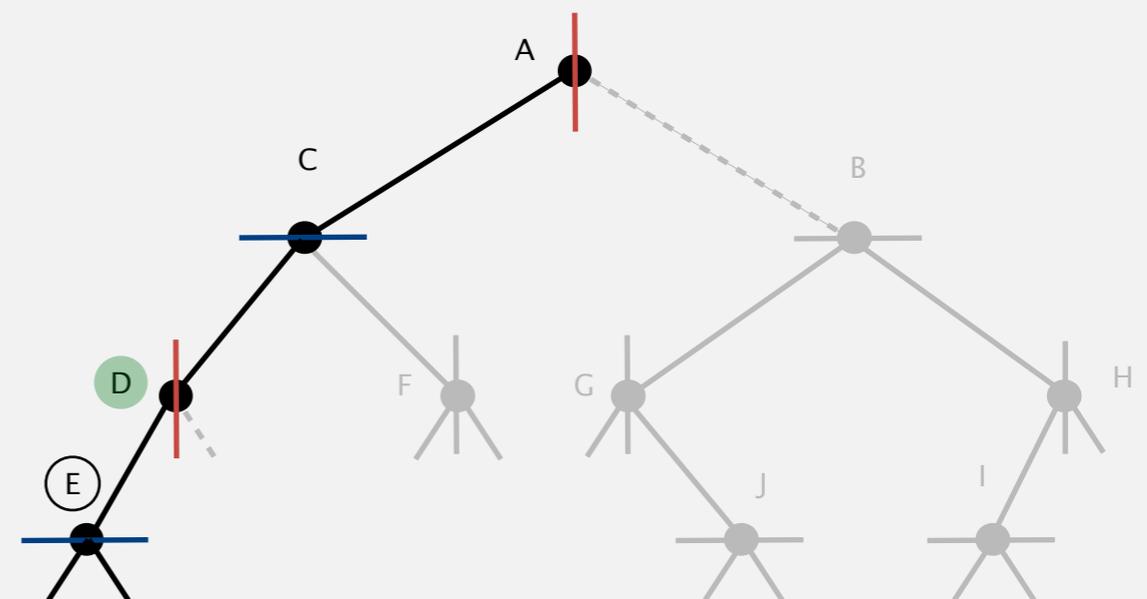
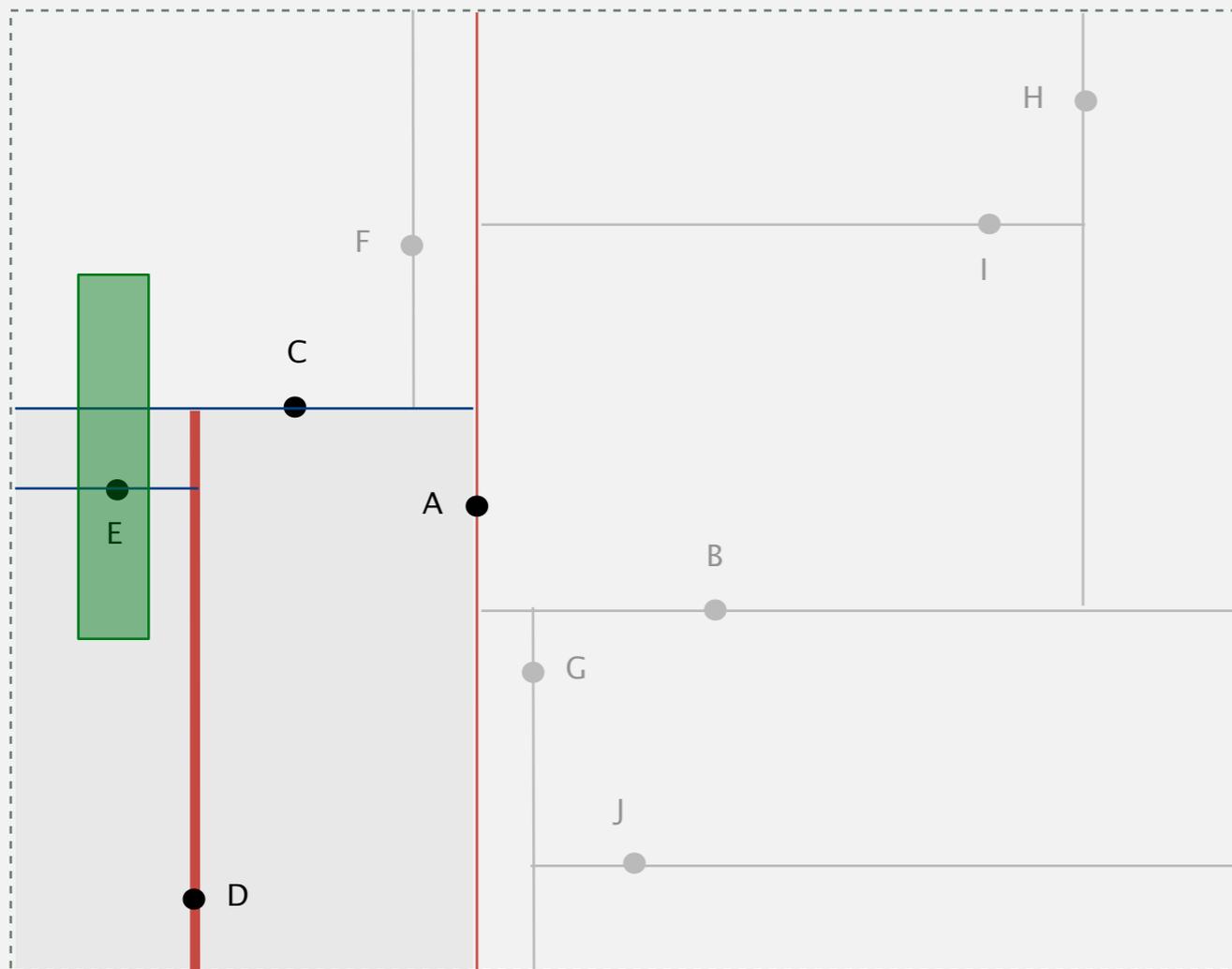


return from function call

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

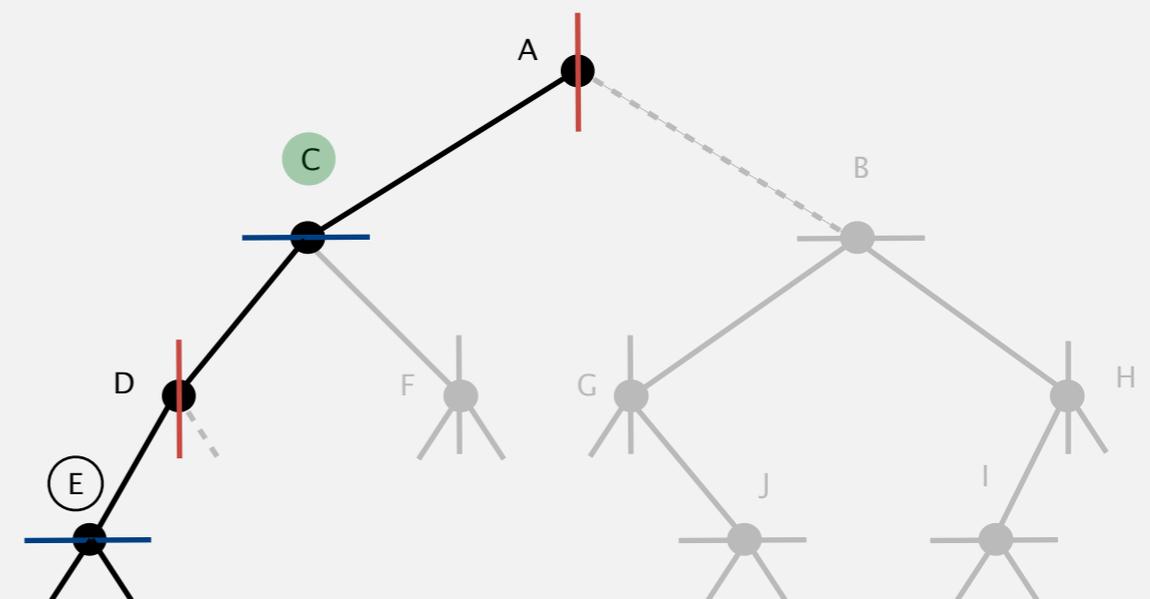
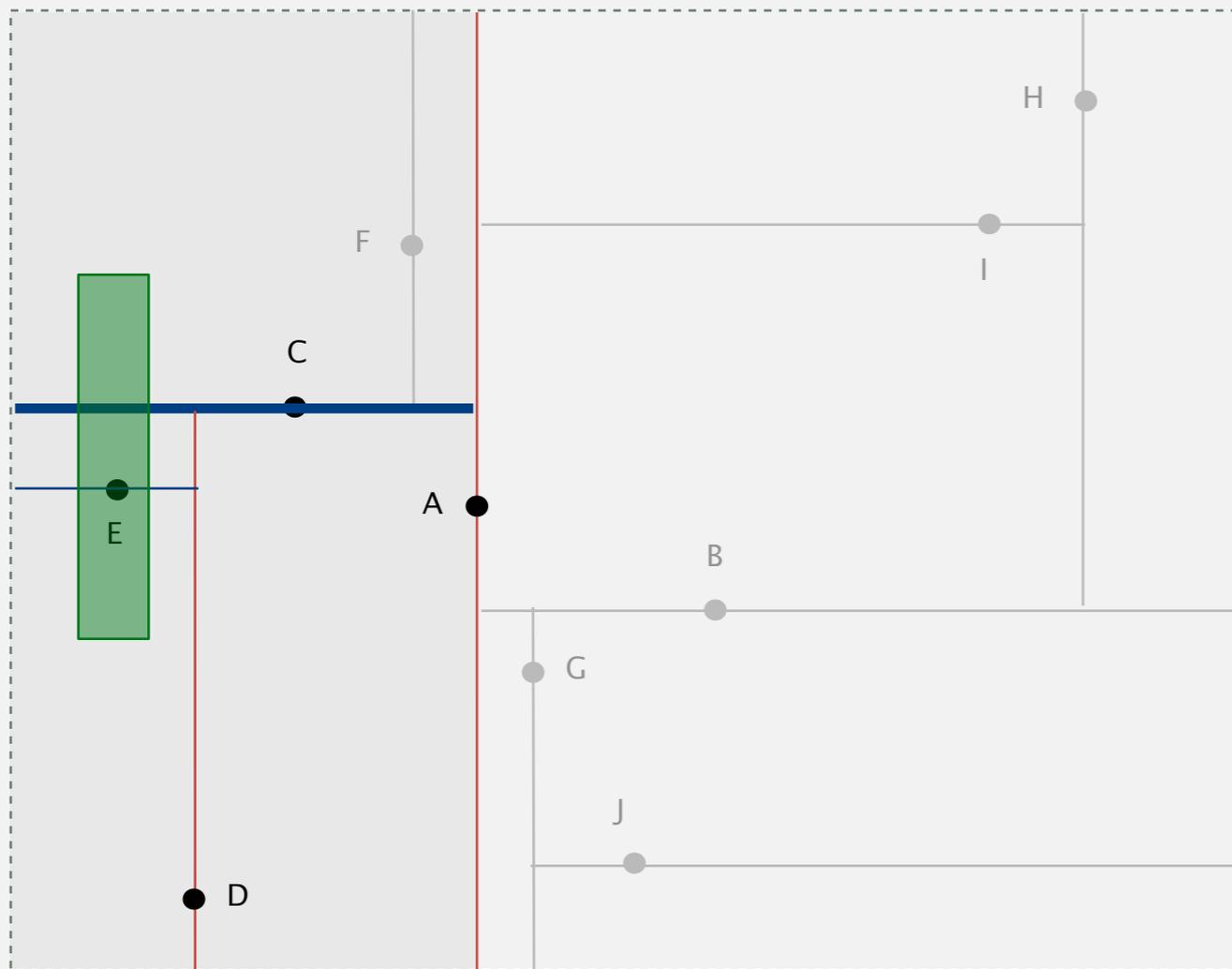


return from function call

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

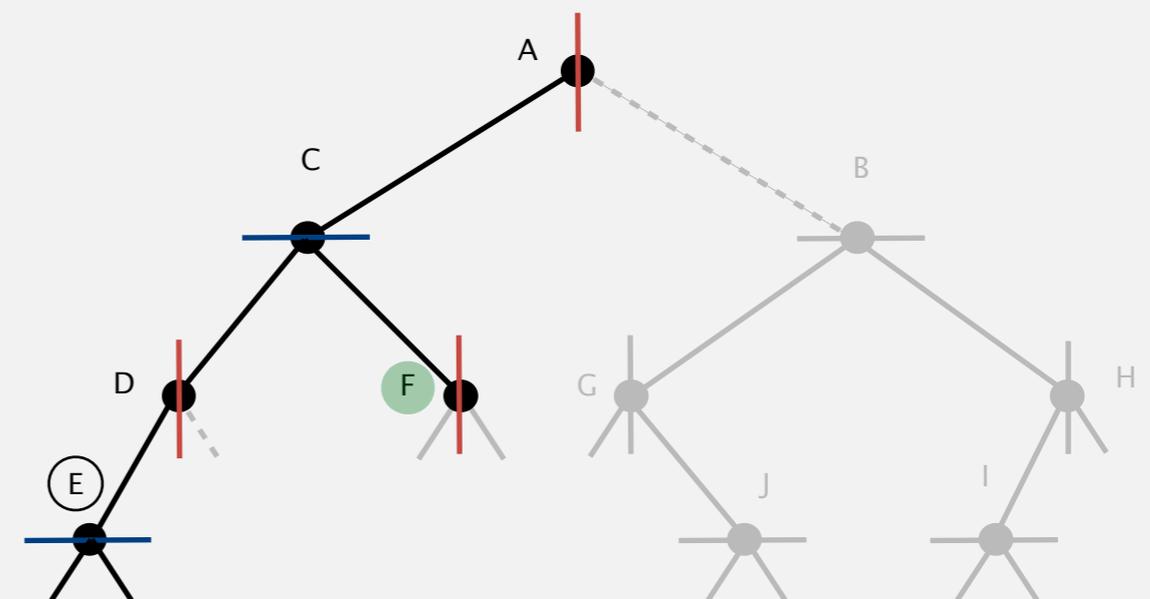
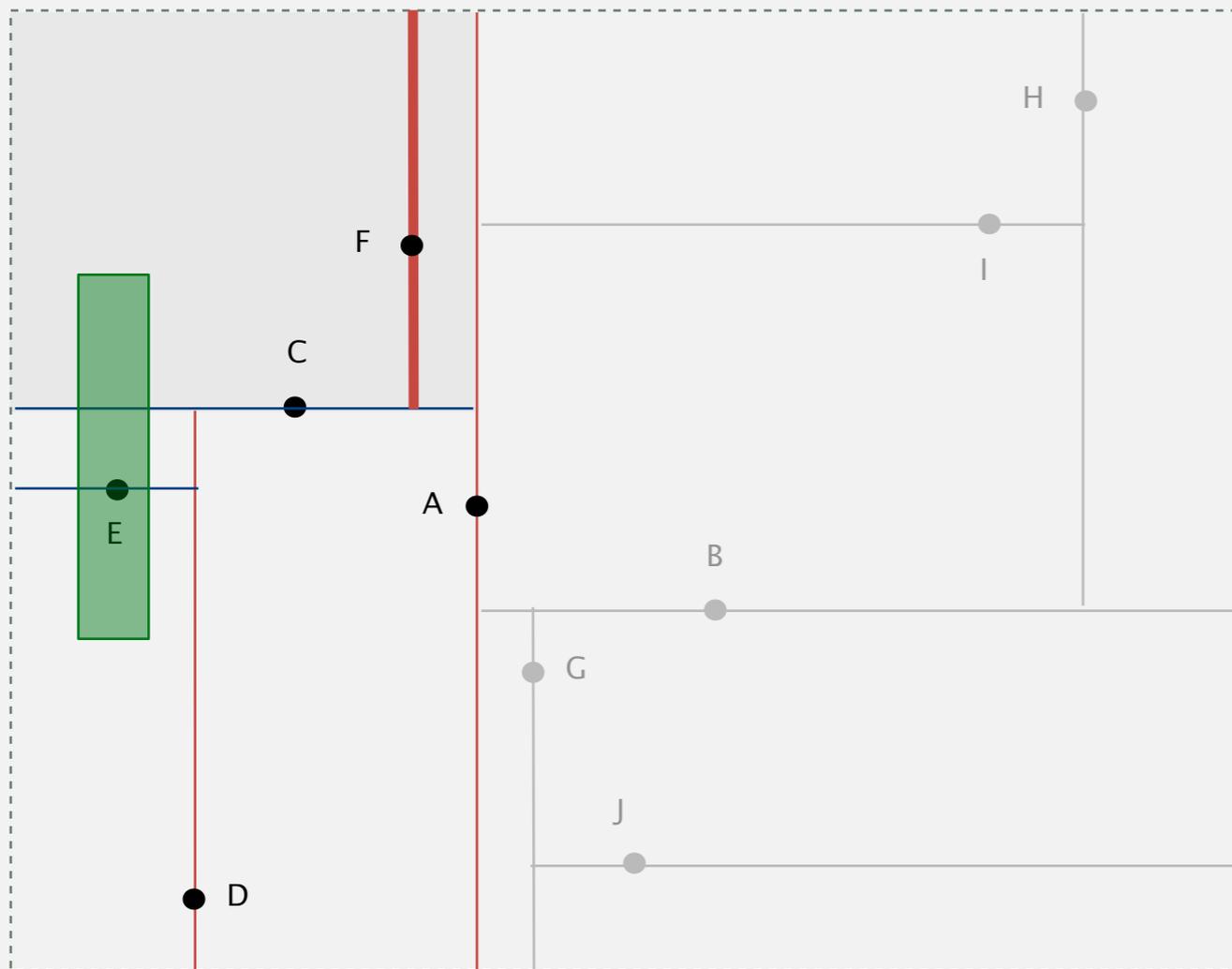


return from function call

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

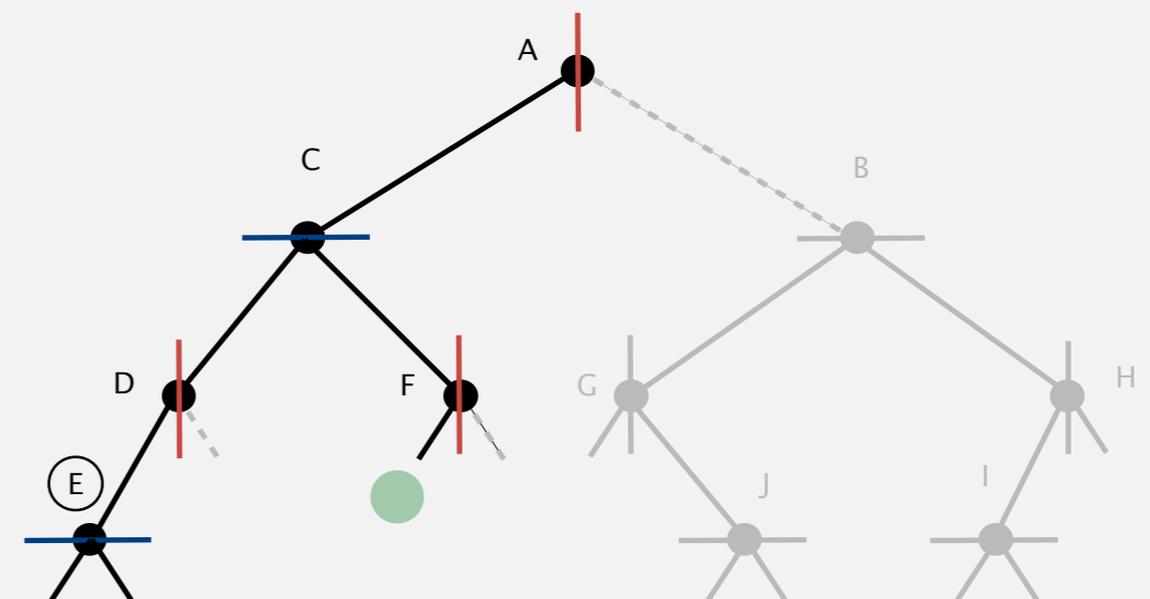
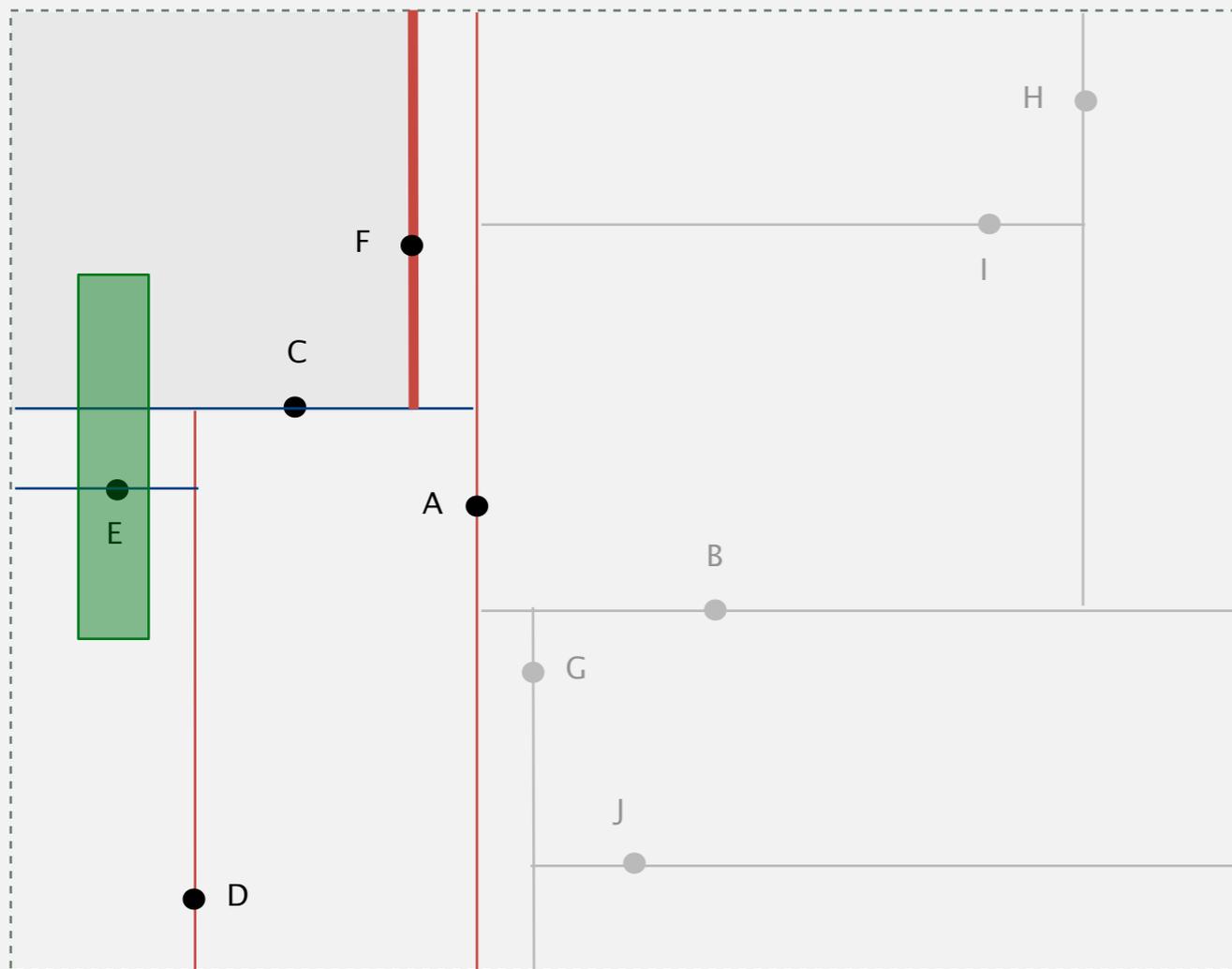


**query rectangle to left of splitting line
search only in left subtree**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

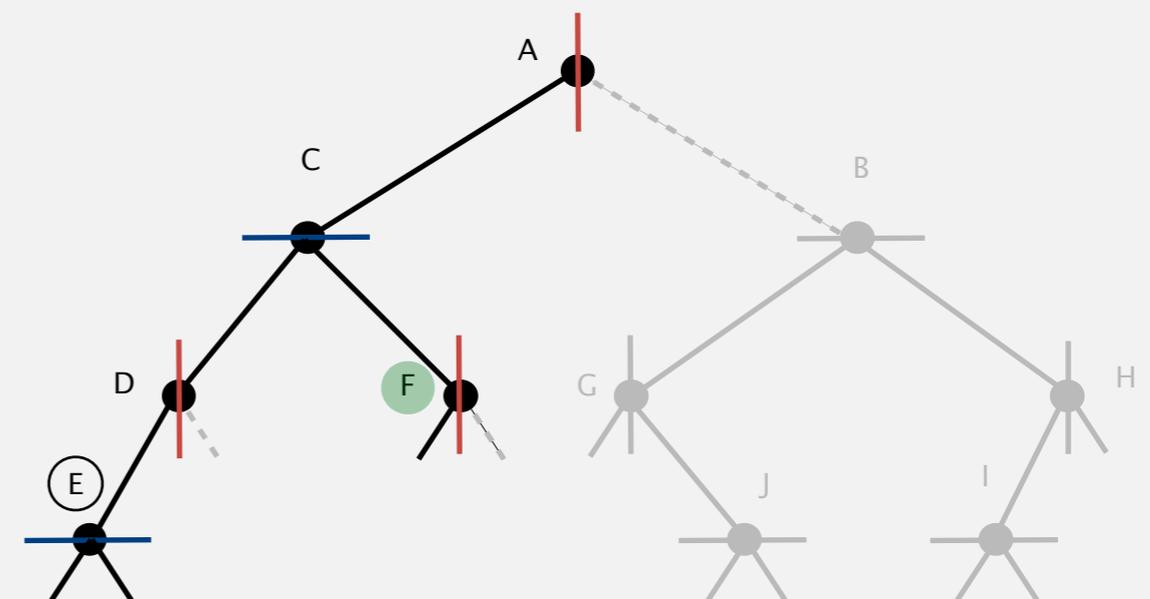
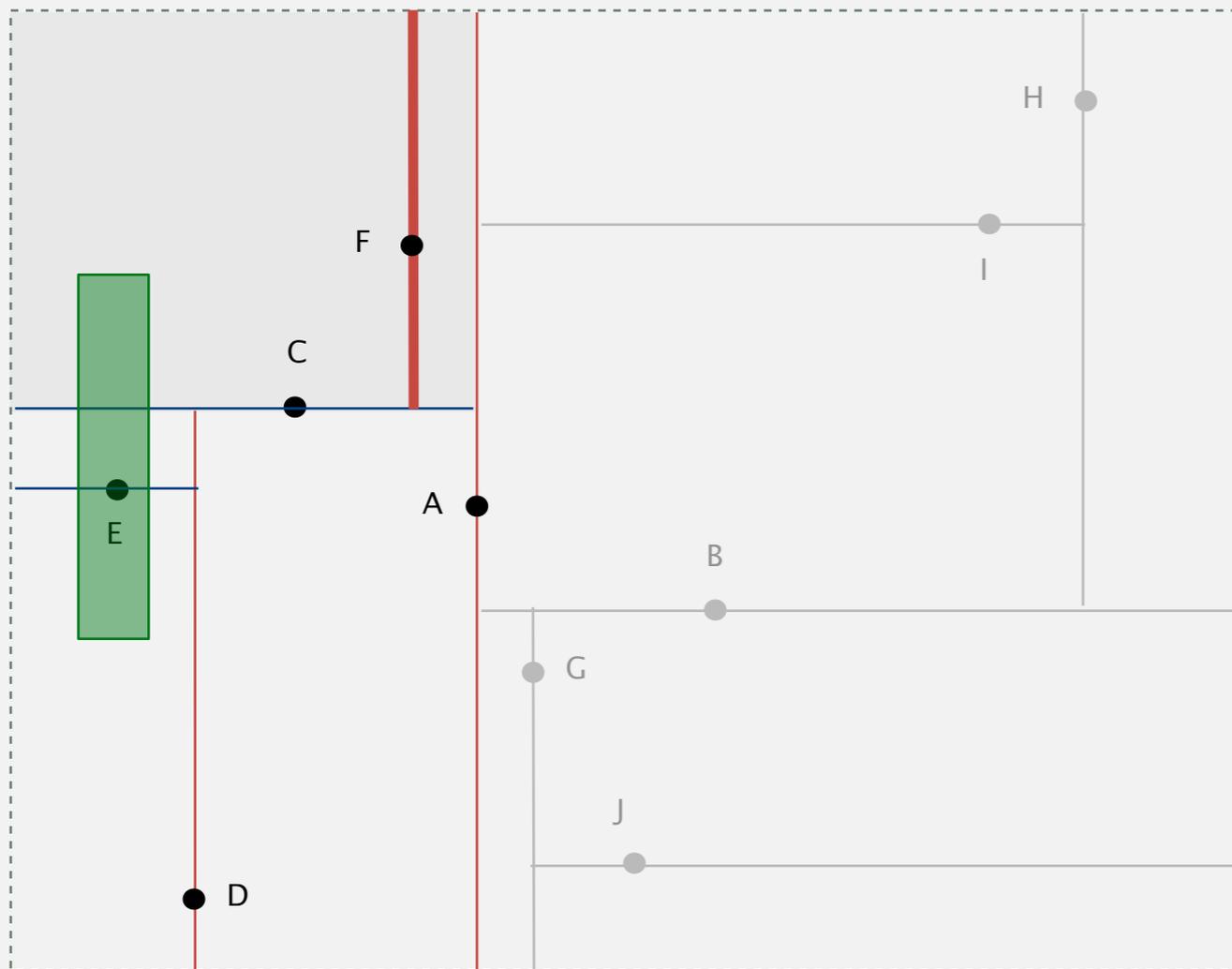


**search left subtree
stop since empty**

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

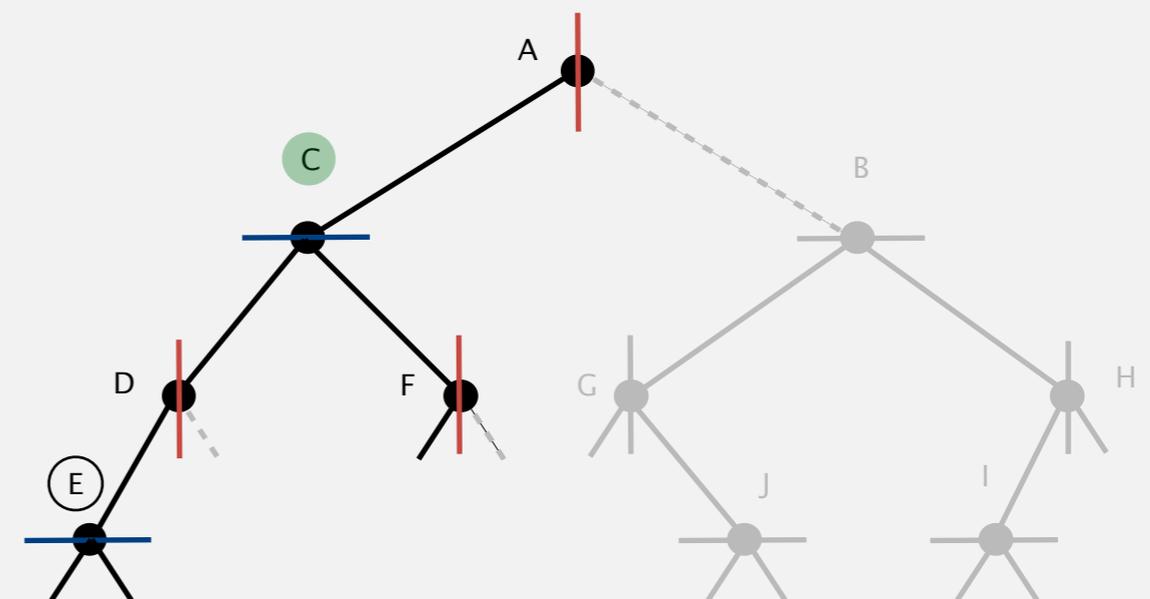
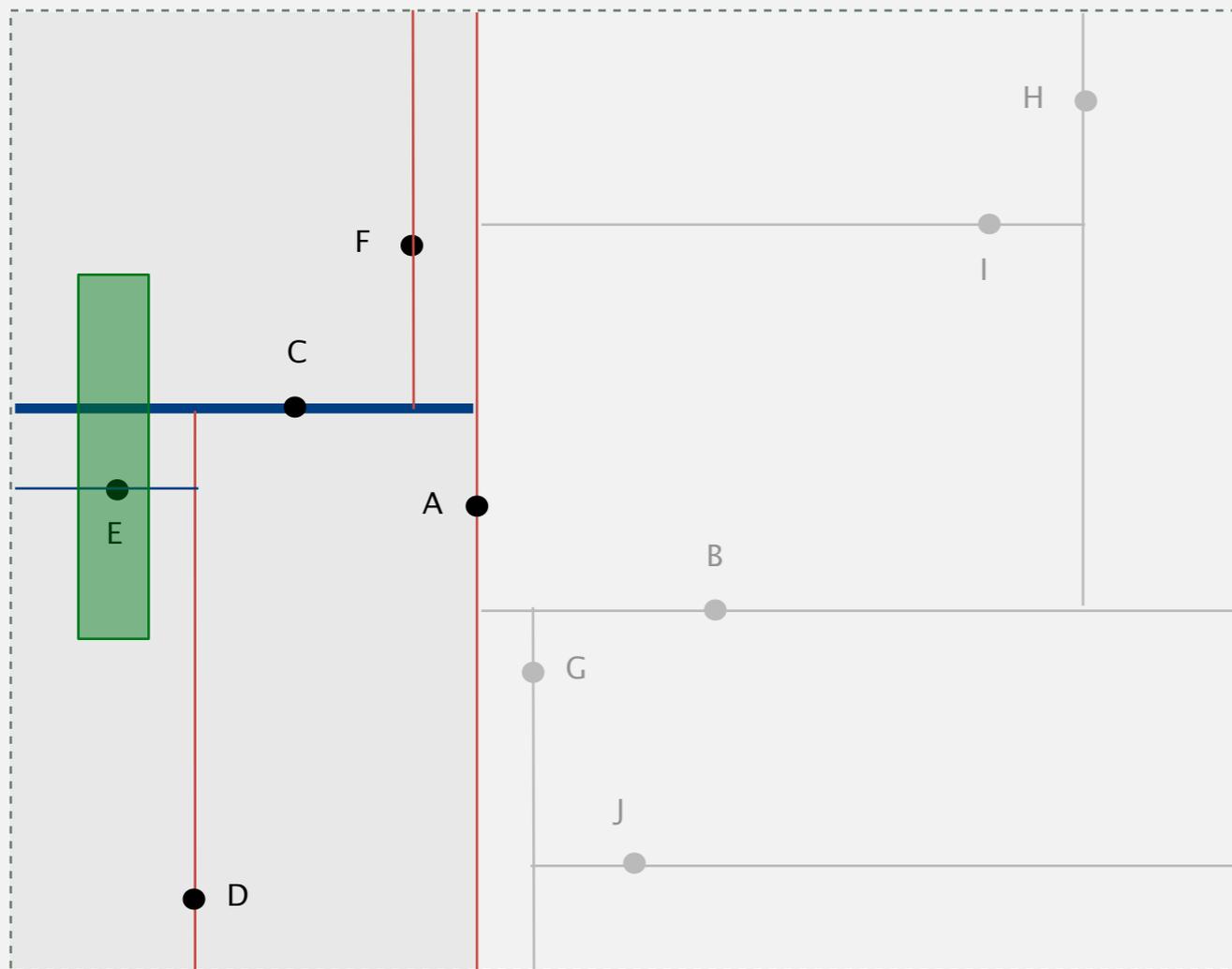


return from function call

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).

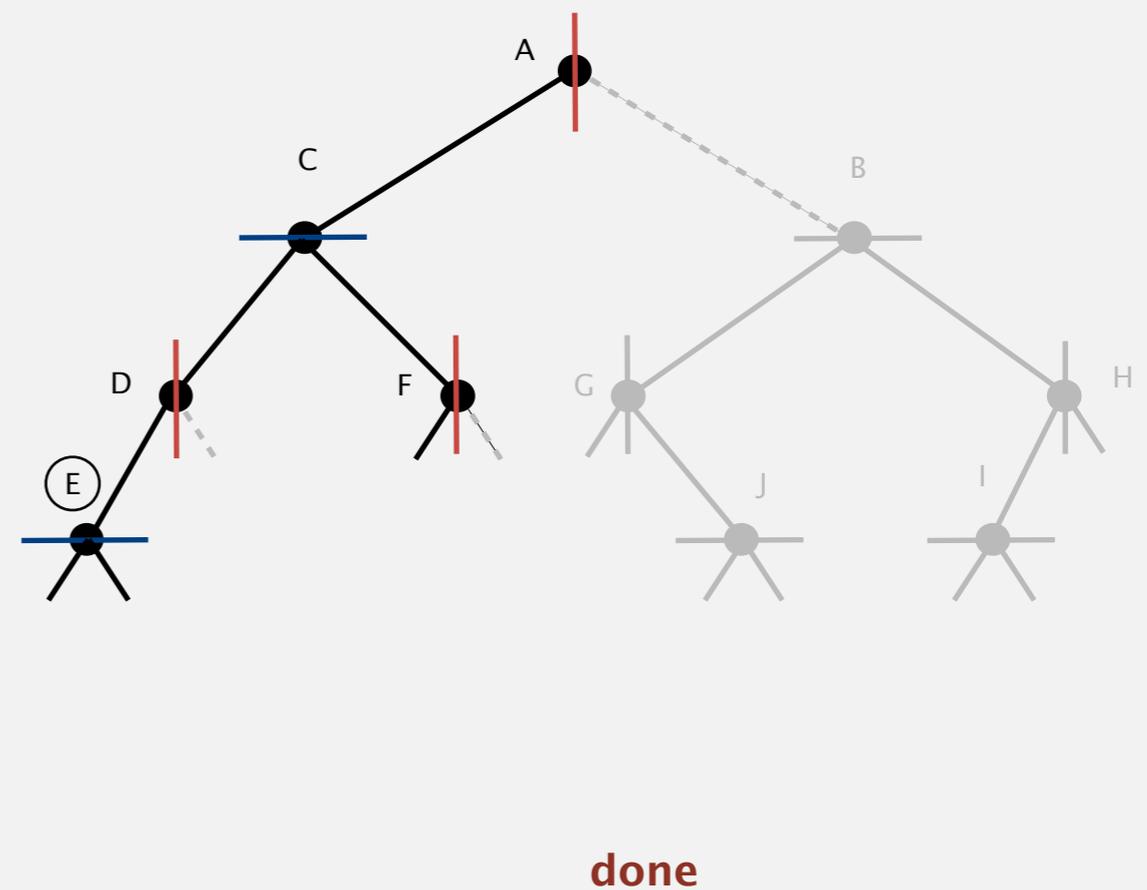
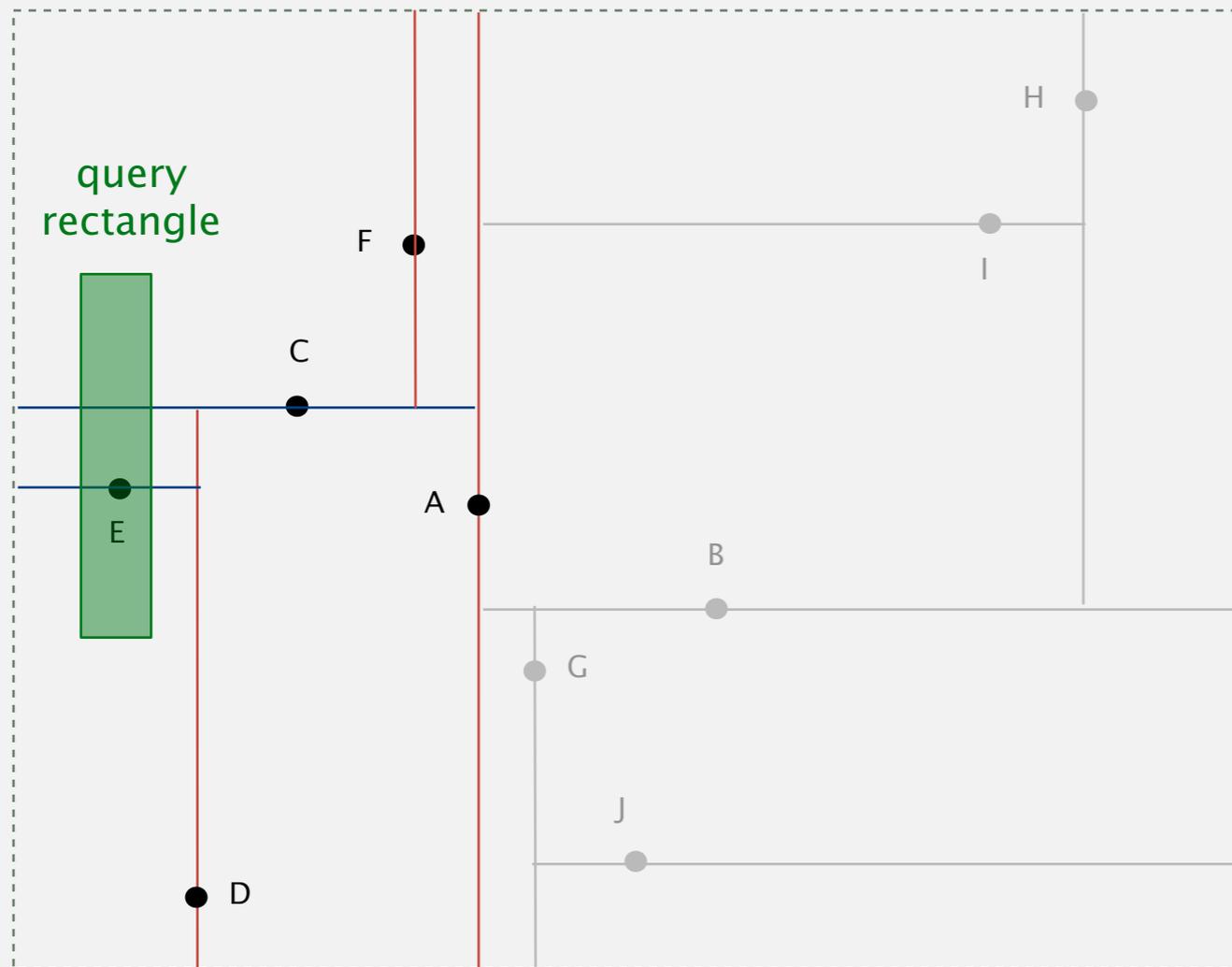


return from function call

2d tree demo: range search

Goal. Find all points in a query axis-aligned rectangle.

- Check if point in node lies in given rectangle.
- Recursively search left/bottom (if any could fall in rectangle).
- Recursively search right/top (if any could fall in rectangle).





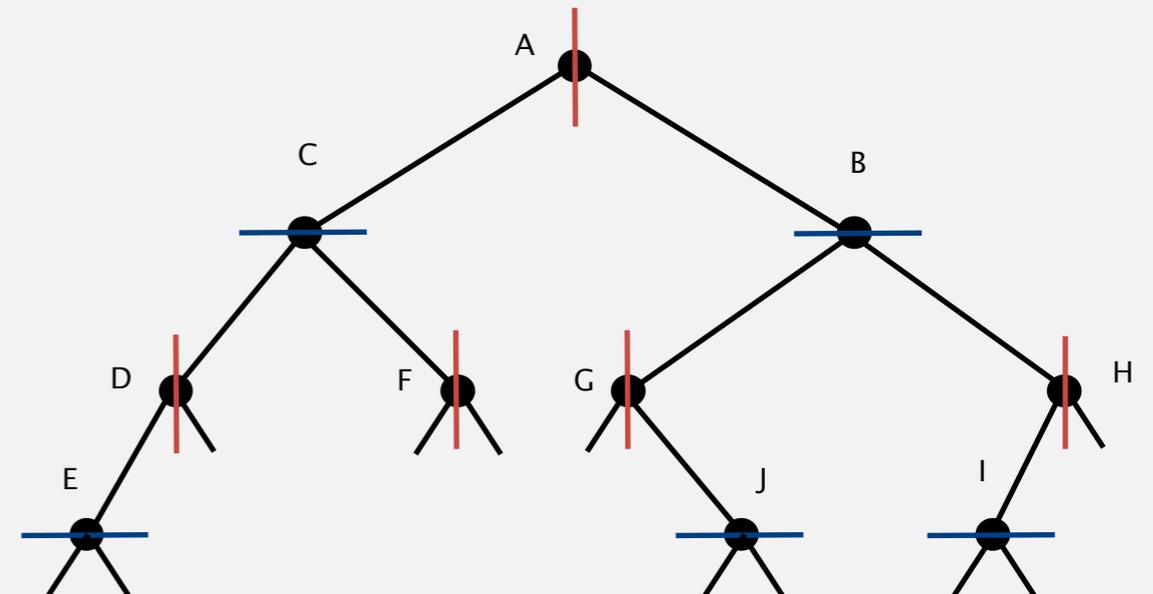
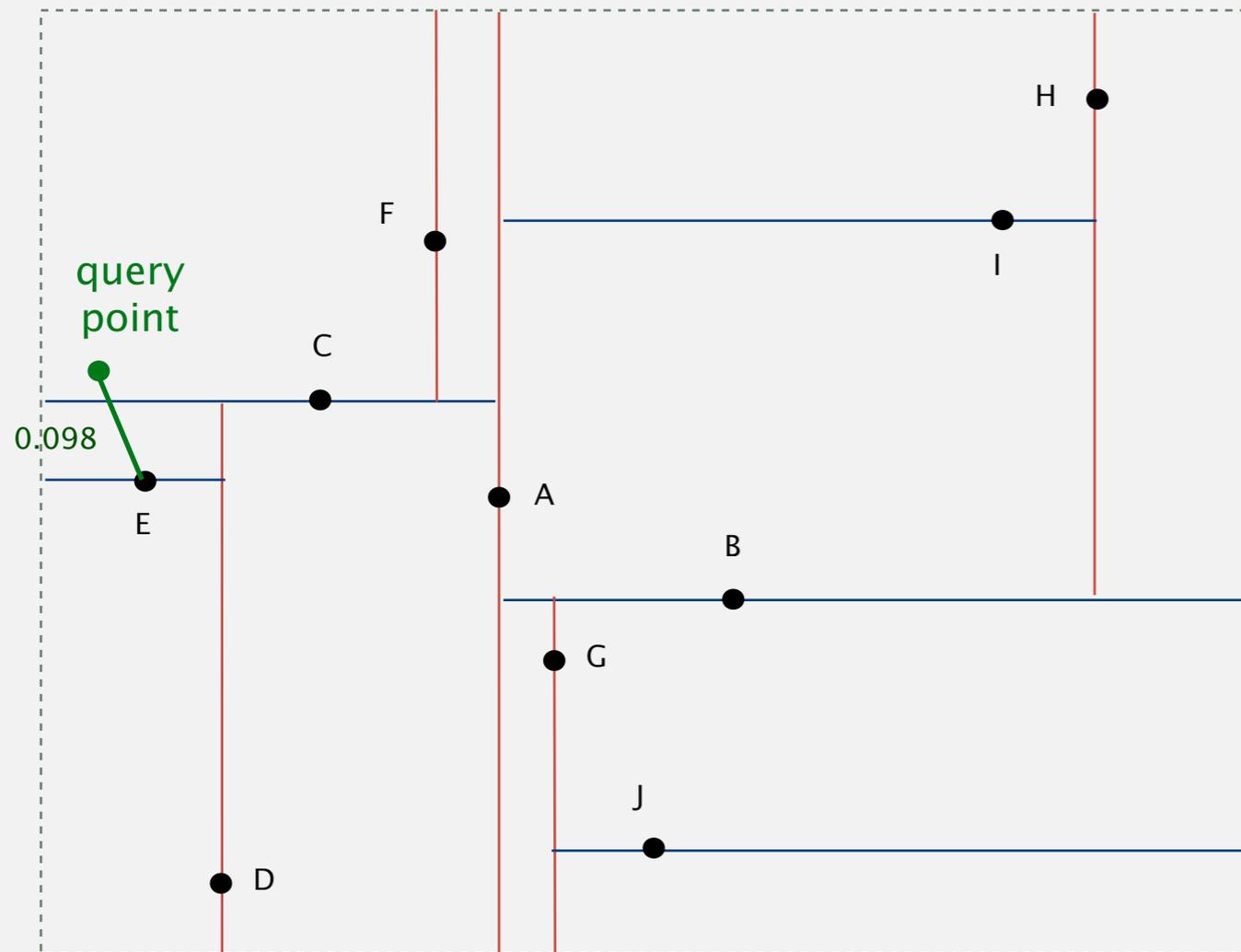
<http://algs4.cs.princeton.edu>

2D TREE DEMO

- ▶ *insertion*
- ▶ *range search*
- ▶ *nearest neighbor*

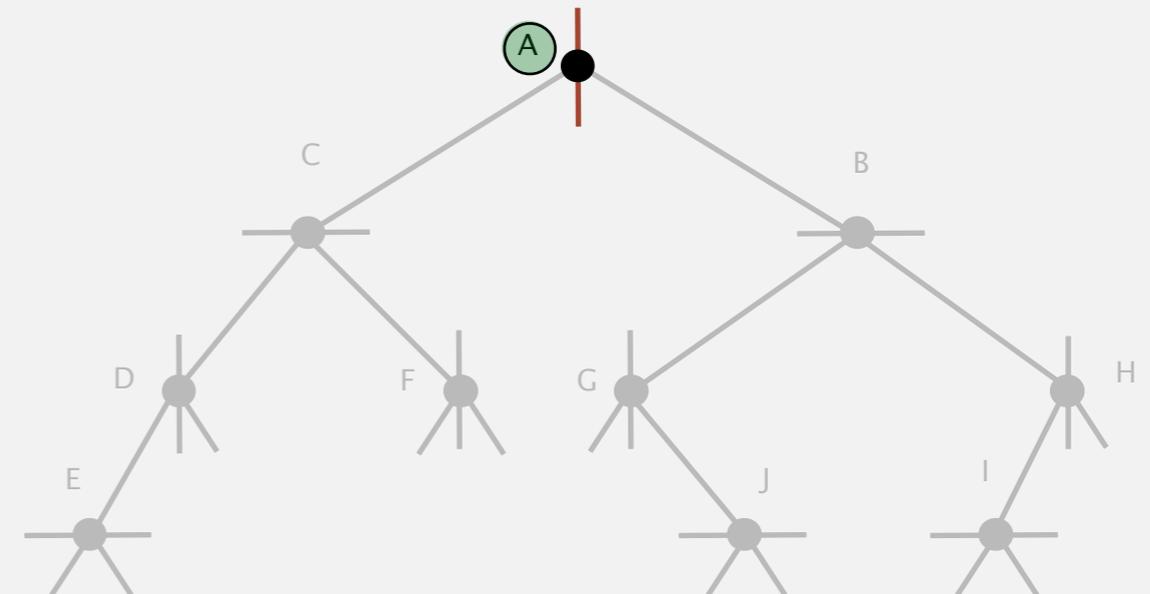
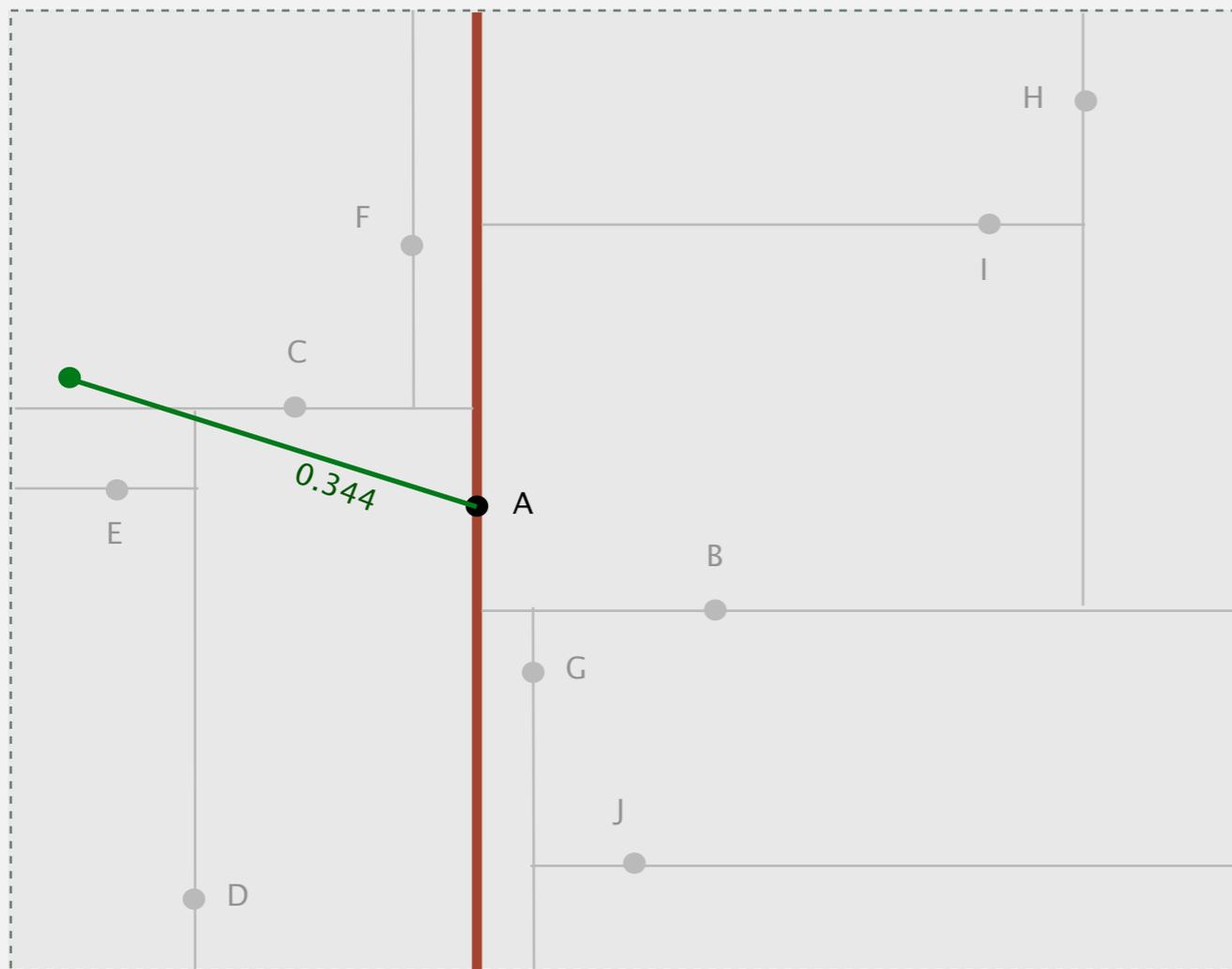
2d tree demo: nearest neighbor

Goal. Find closest point to query point.



2d tree demo: nearest neighbor

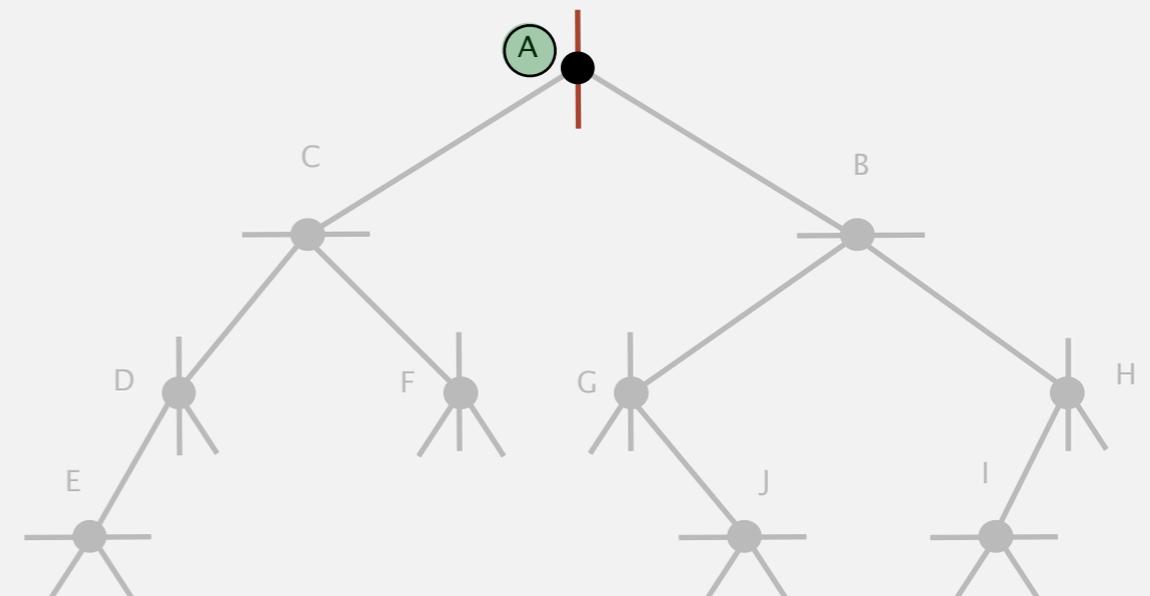
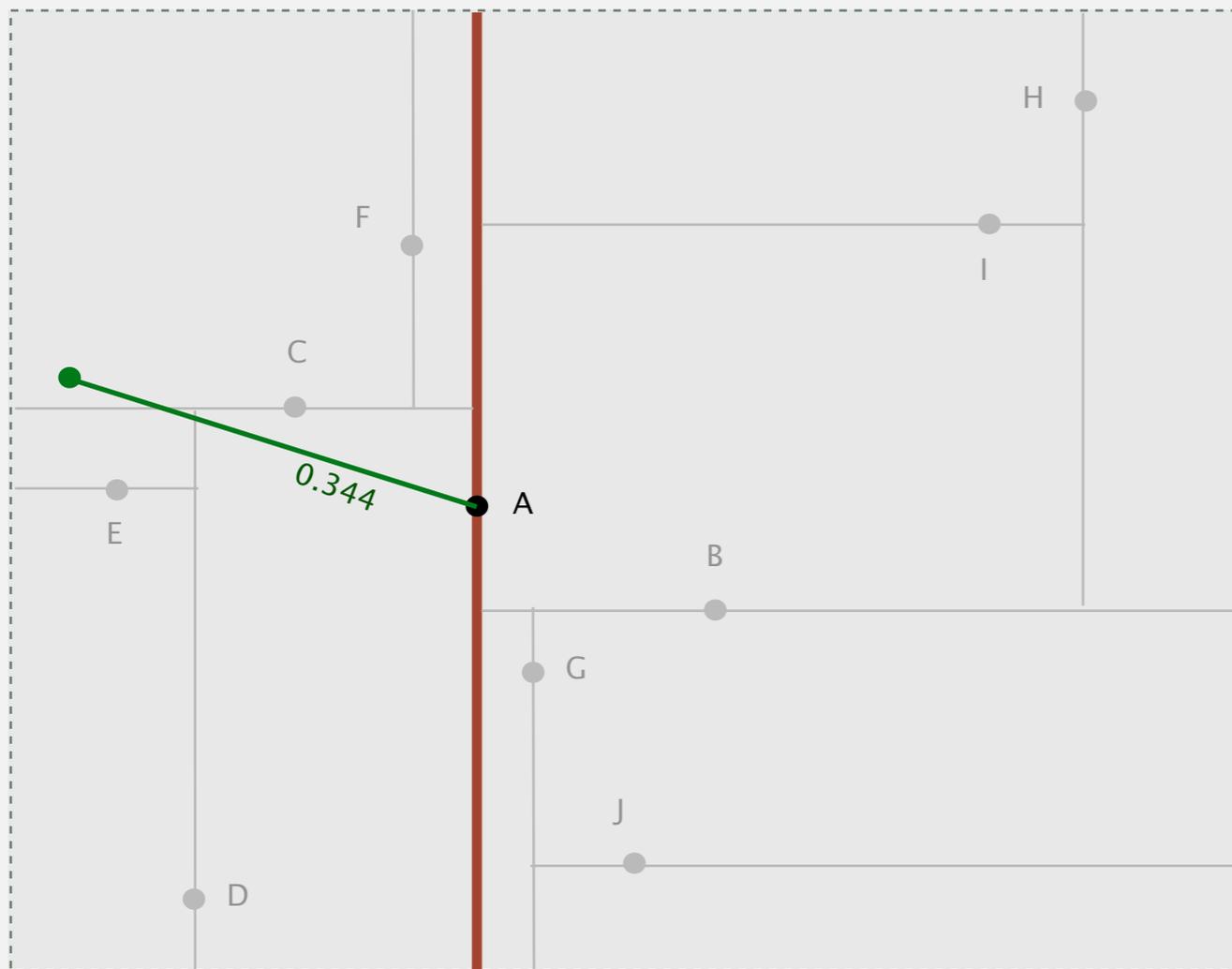
- Check distance from point in node to query point.



search root node
compute distance from query point to A
(update champion nearest neighbor)

2d tree demo: nearest neighbor

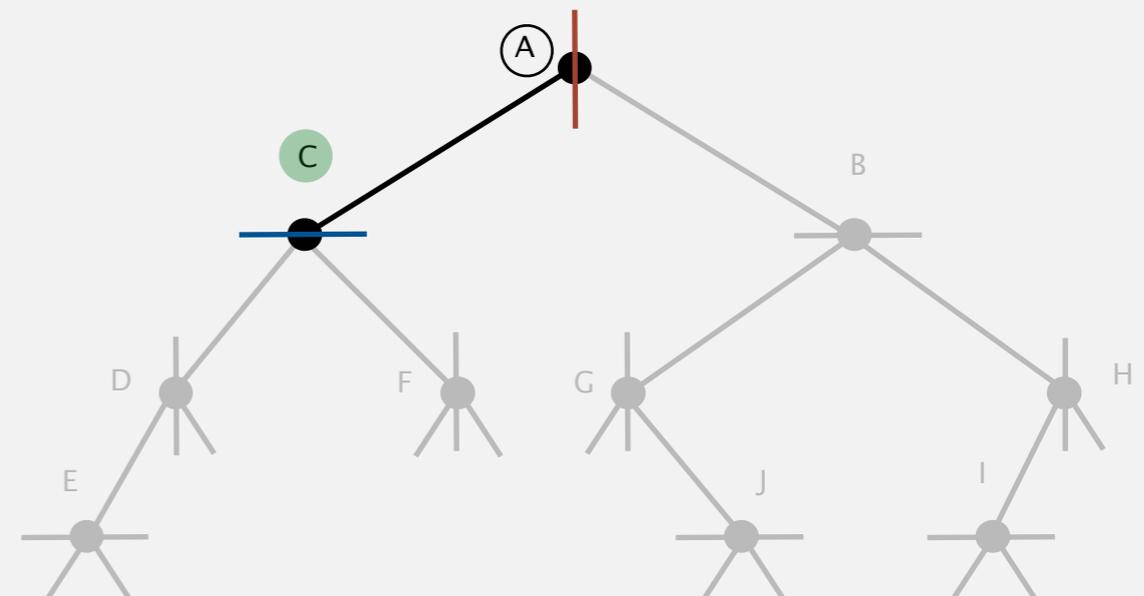
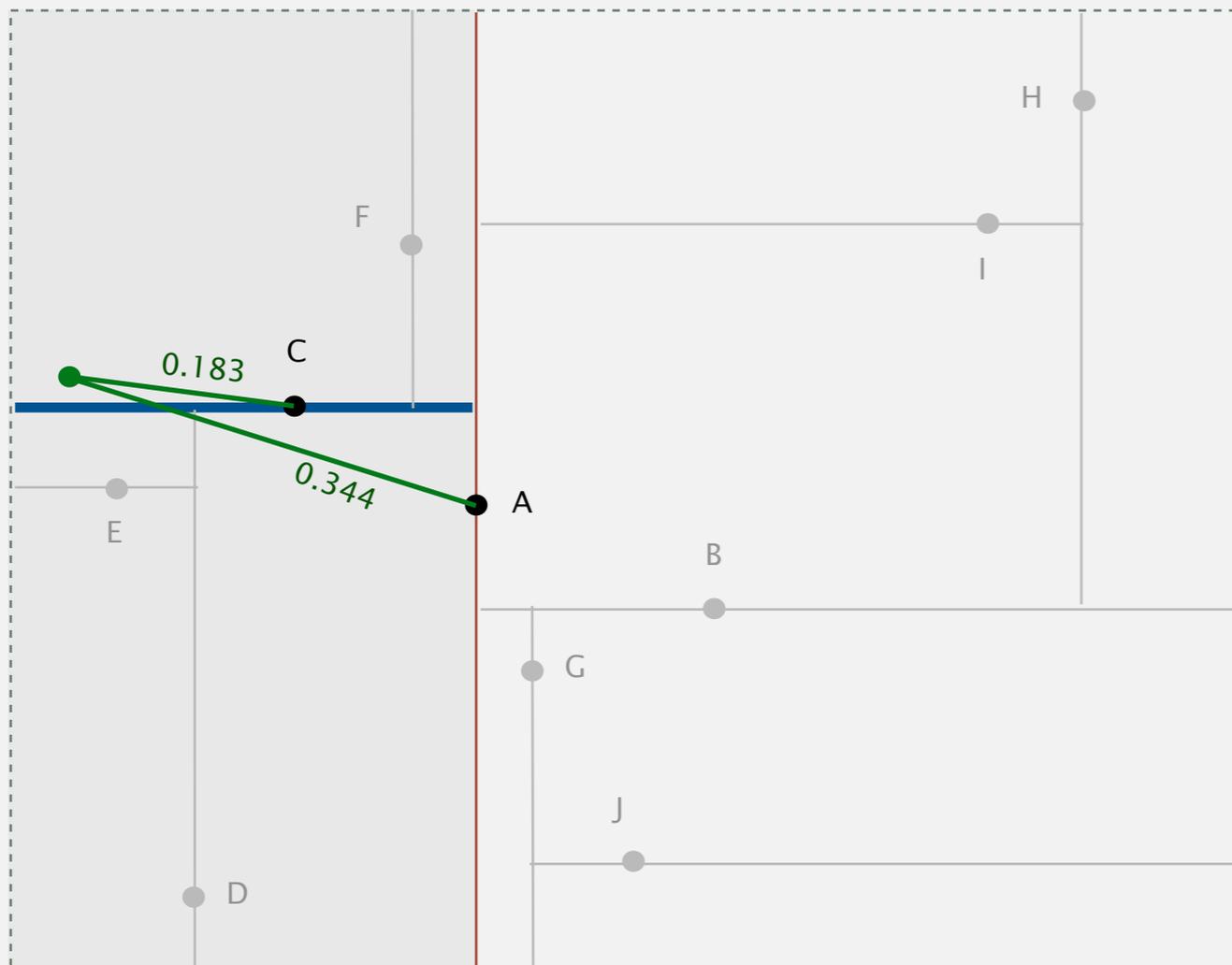
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**query point is to the left of splitting line
search left subtree first**

2d tree demo: nearest neighbor

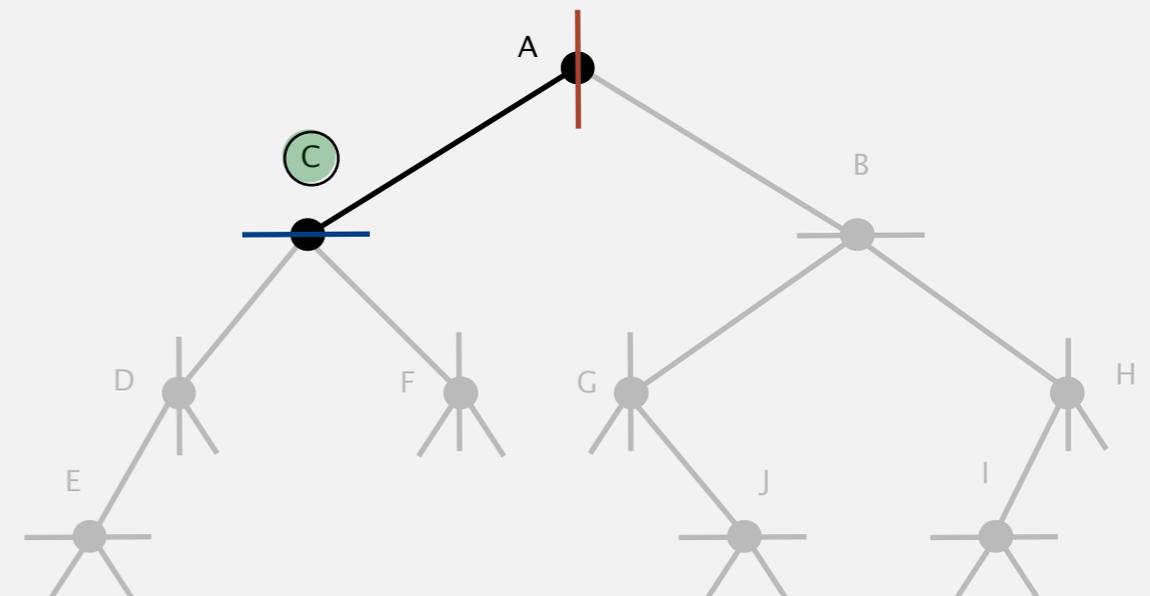
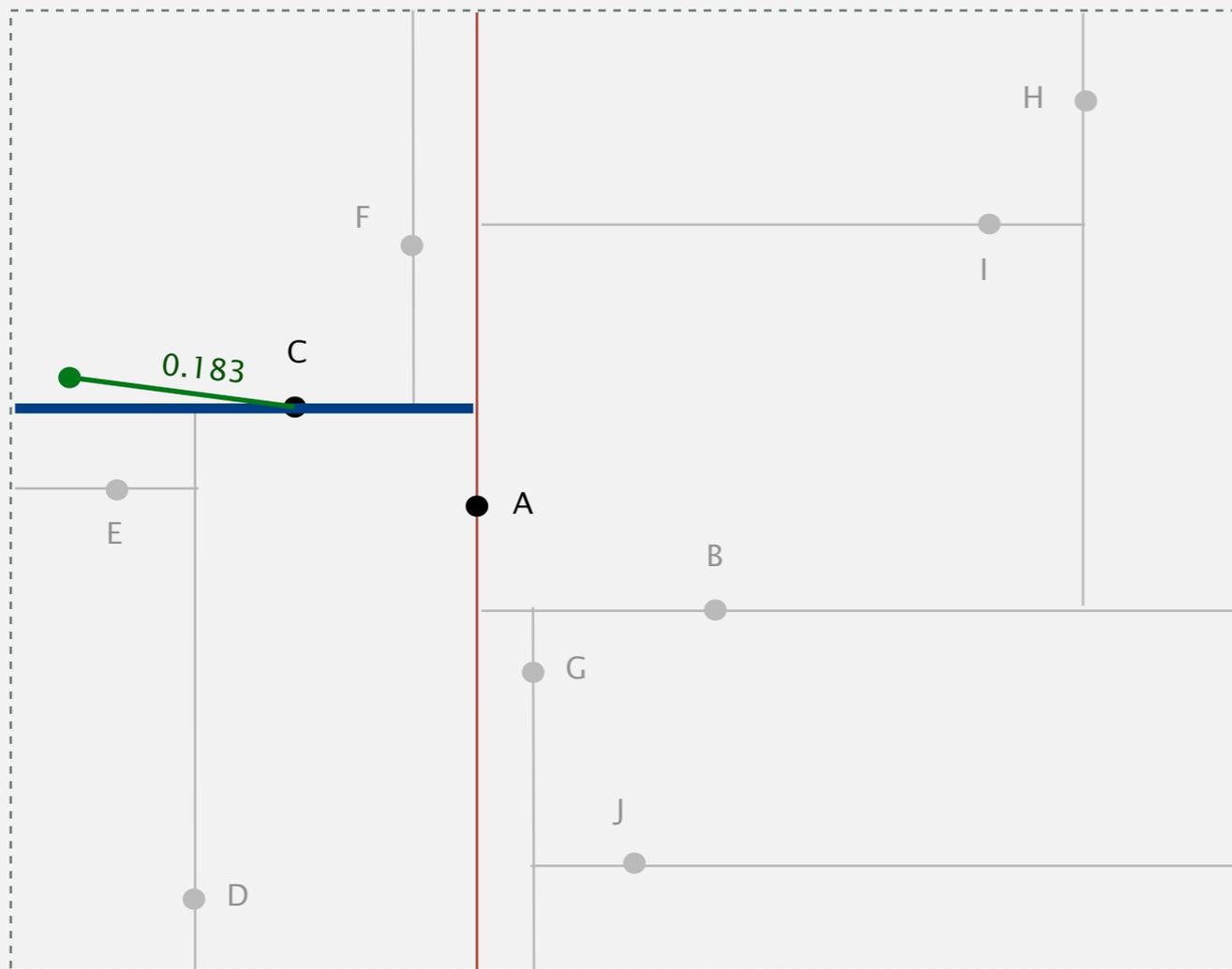
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



search left subtree
compute distance from query point to C
(update champion)

2d tree demo: nearest neighbor

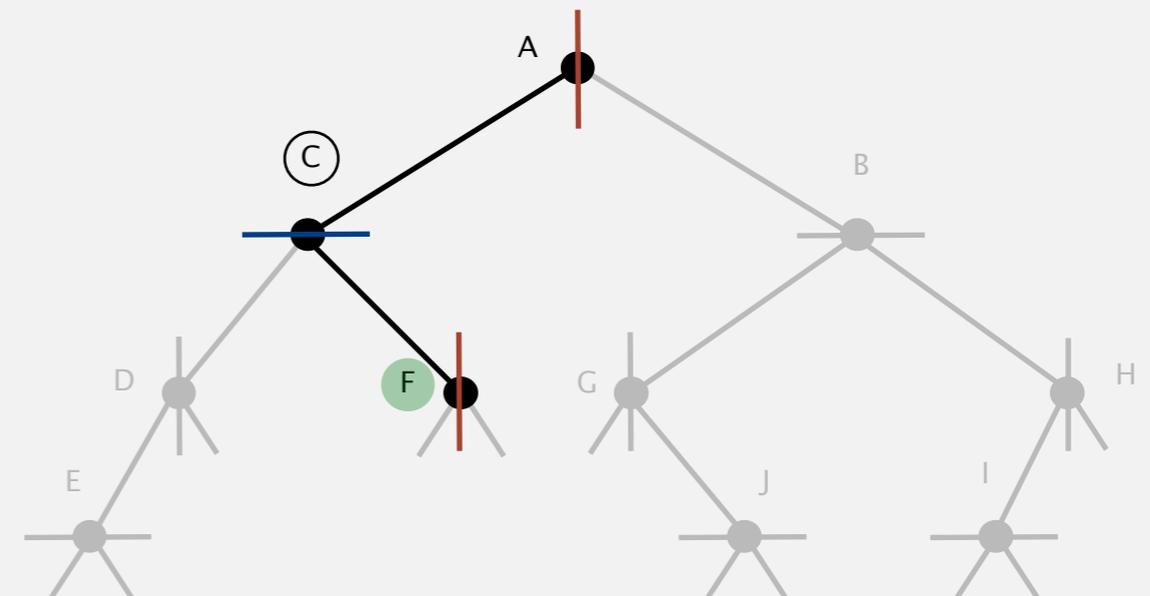
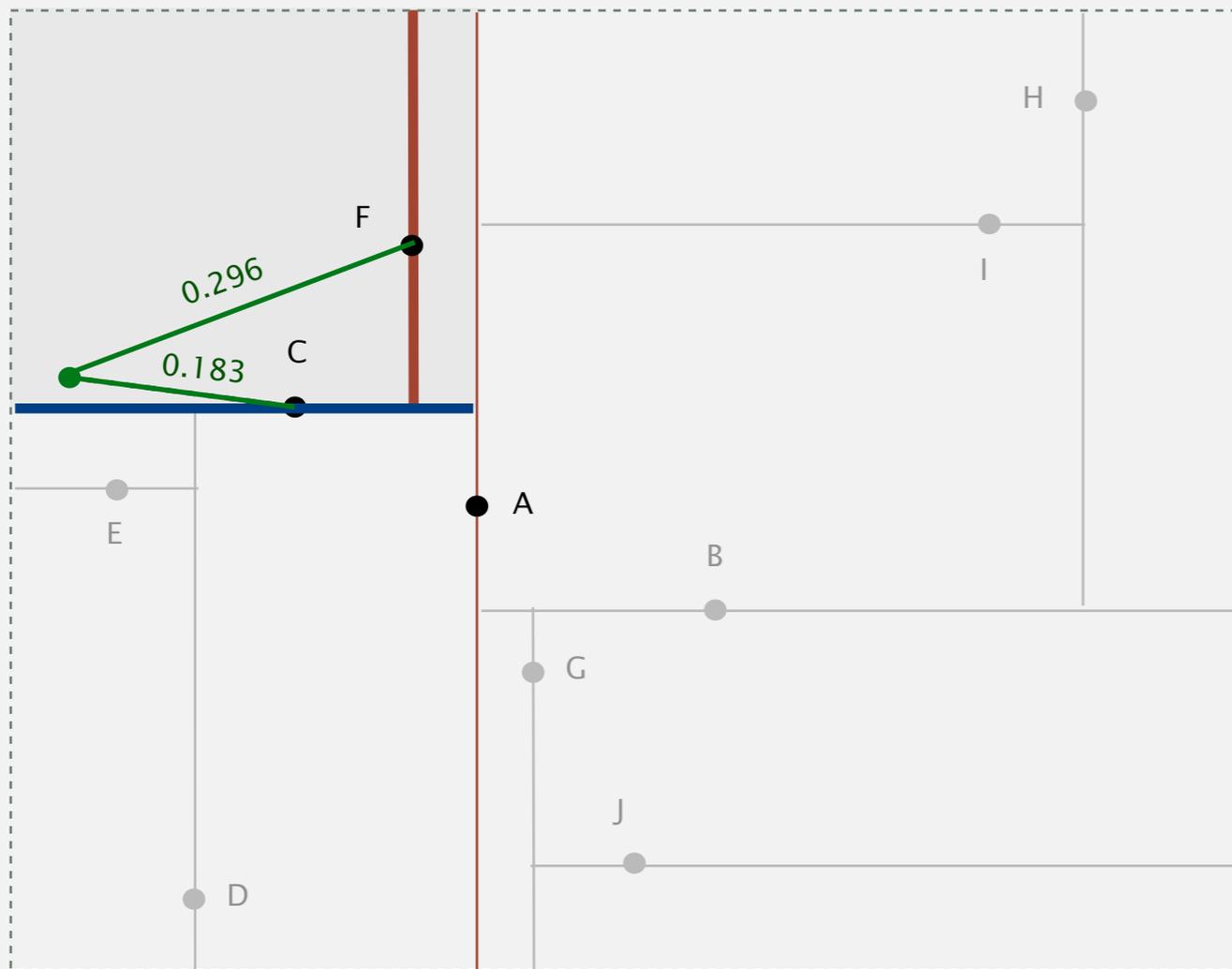
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**query point is above splitting line
search top subtree first**

2d tree demo: nearest neighbor

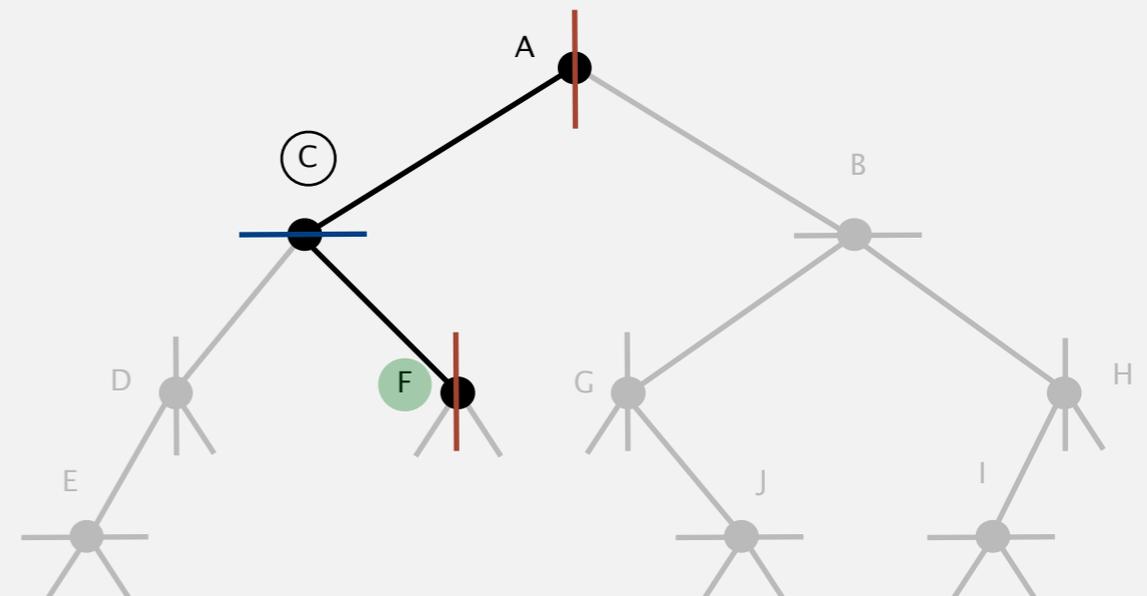
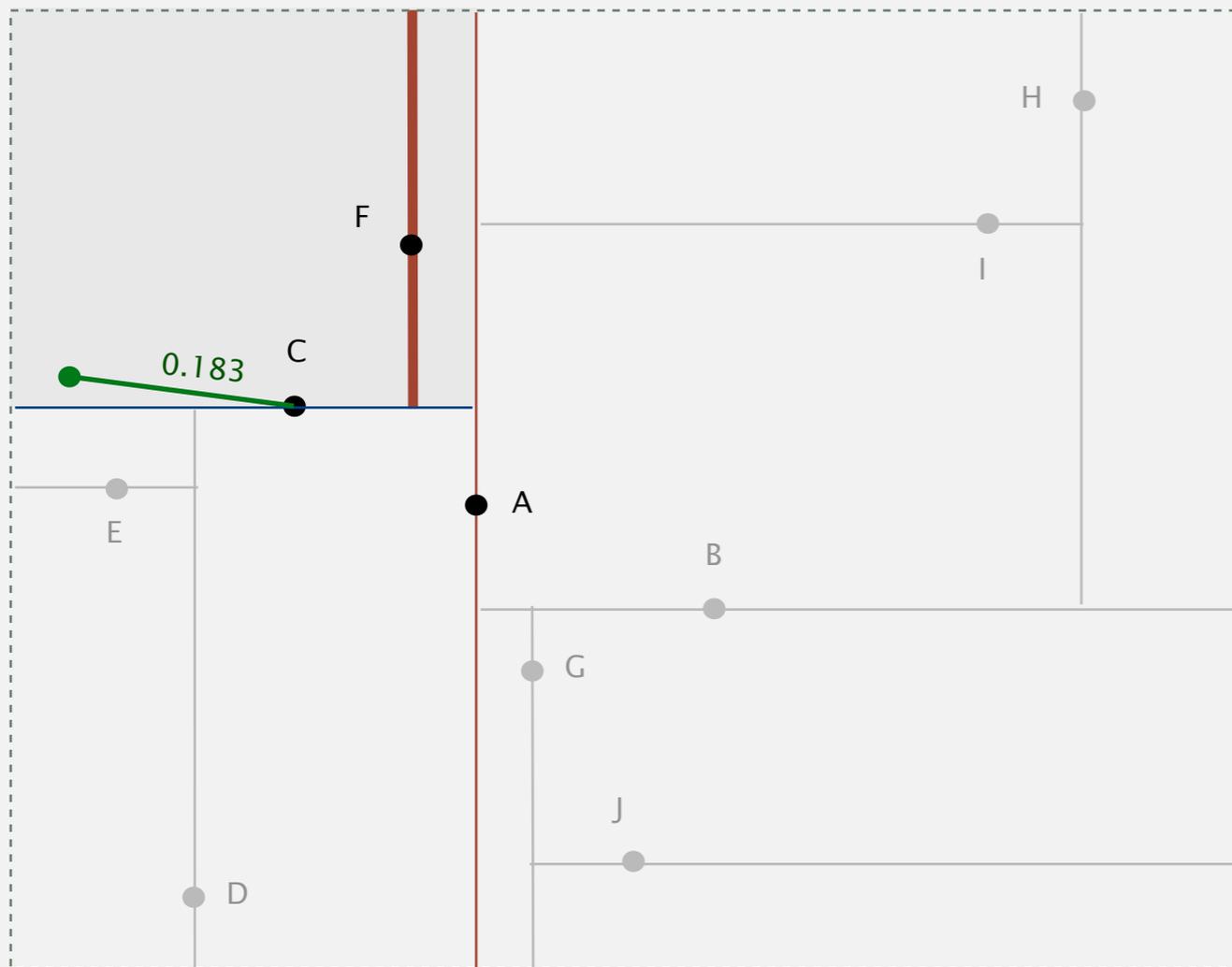
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



search top subtree
compute distance from query point to F

2d tree demo: nearest neighbor

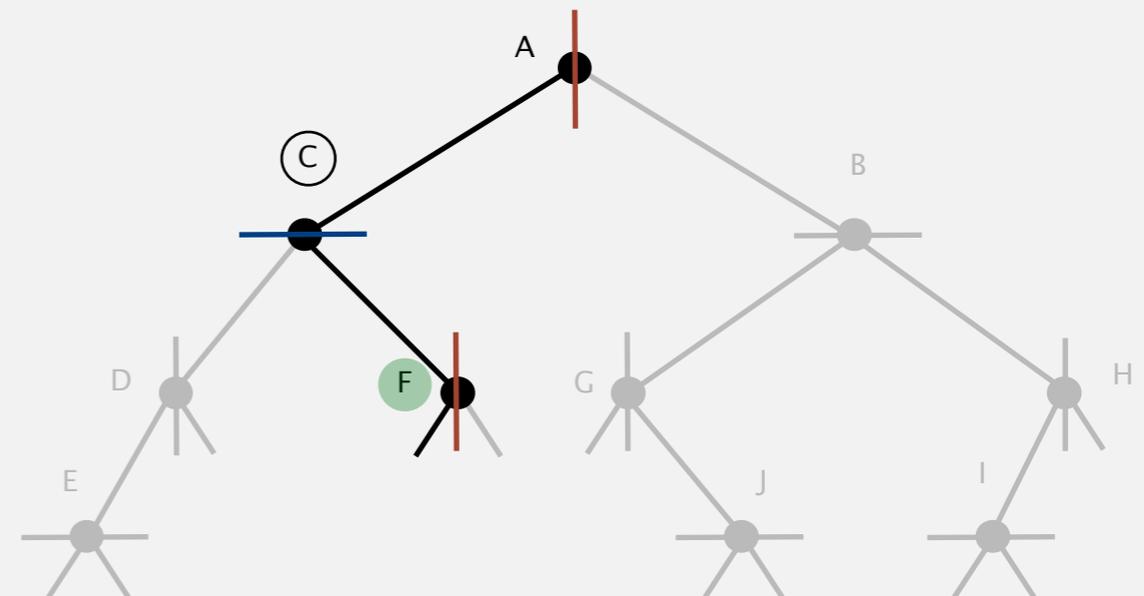
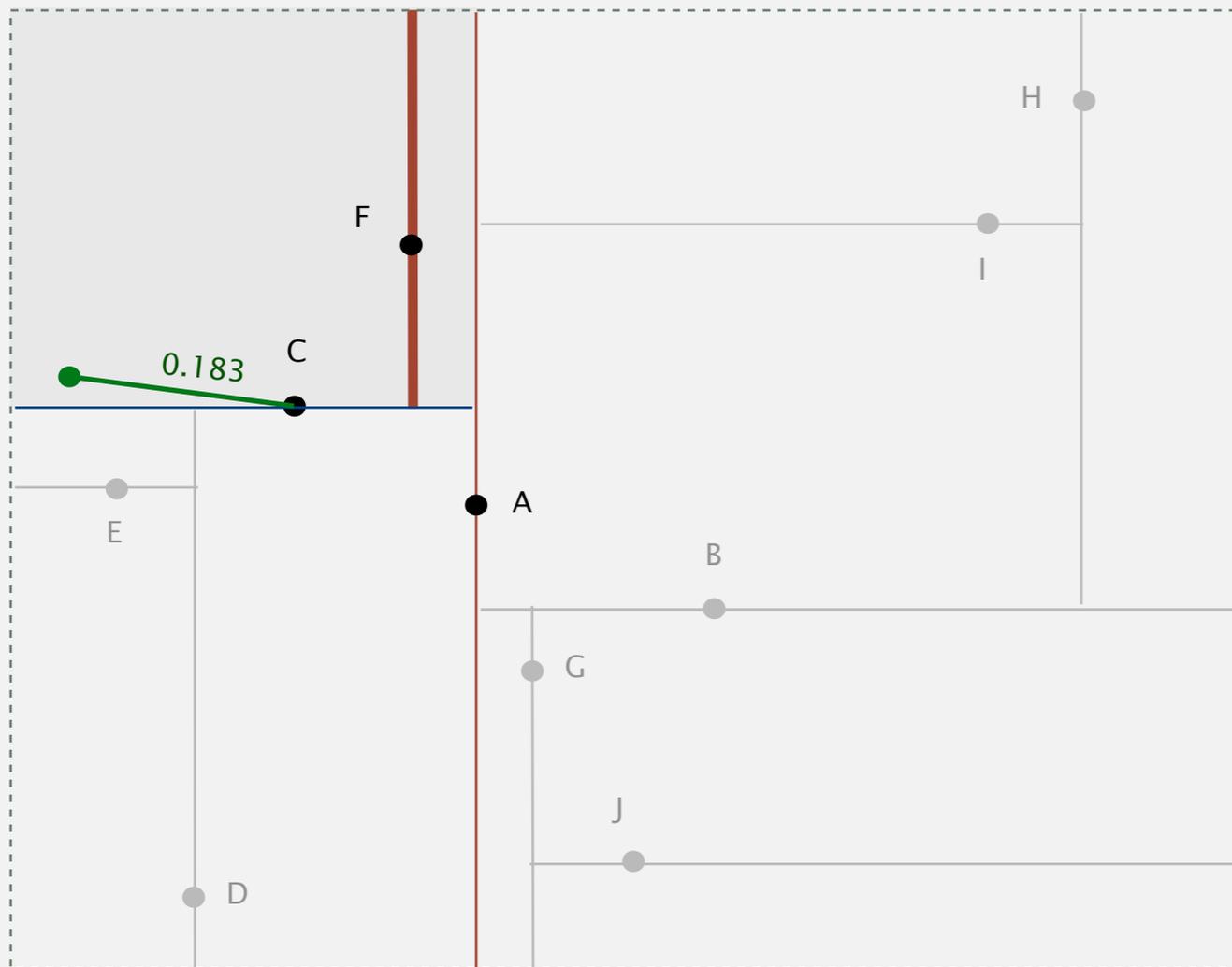
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



query point is to left of splitting line
search left subtree first

2d tree demo: nearest neighbor

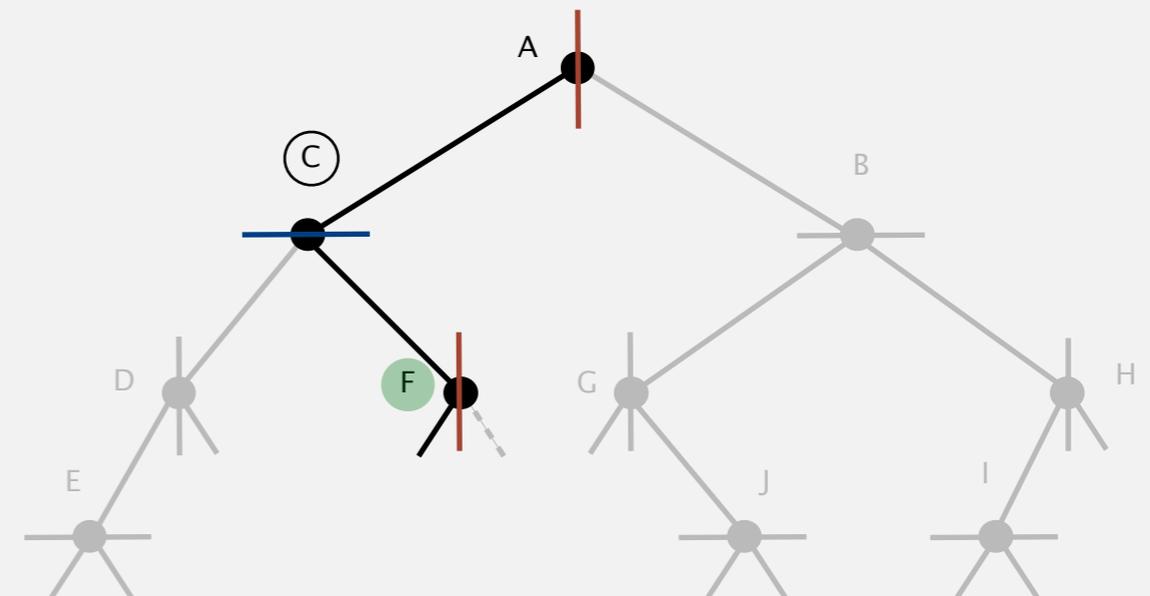
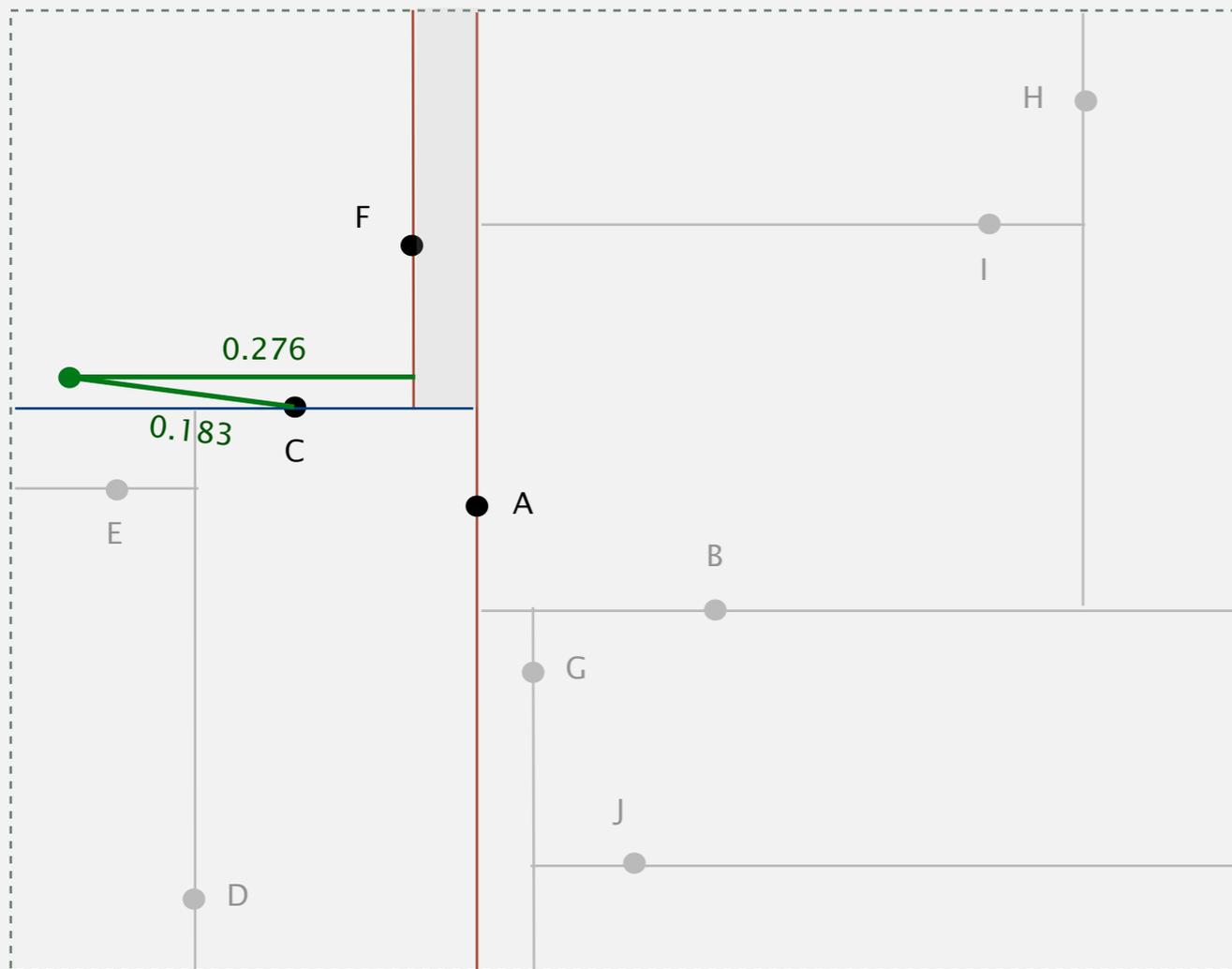
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**search left subtree
return since empty**

2d tree demo: nearest neighbor

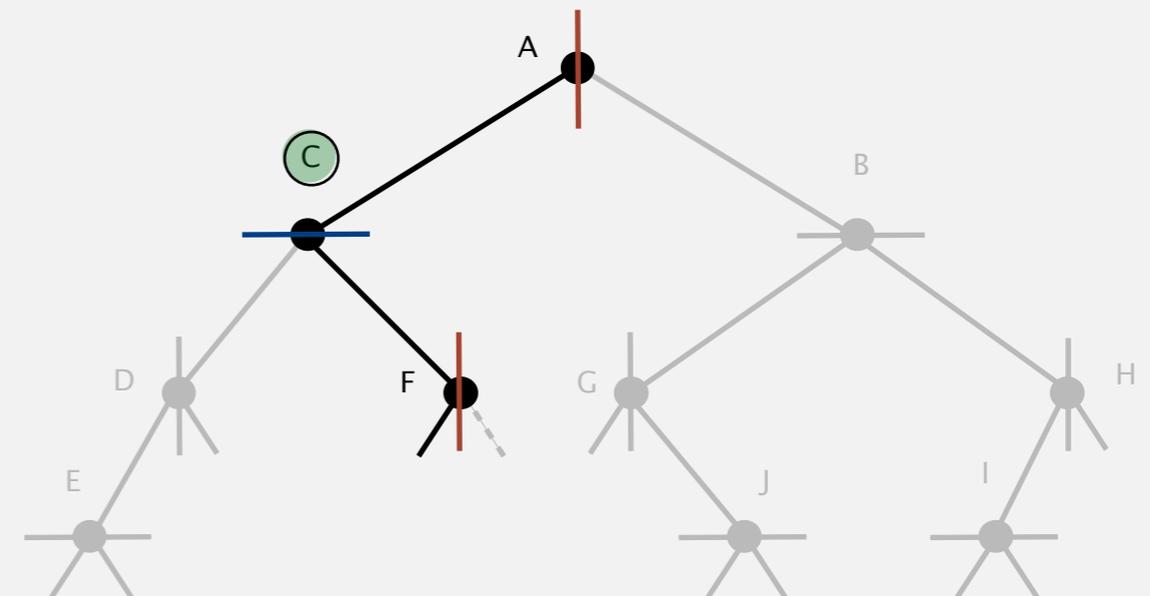
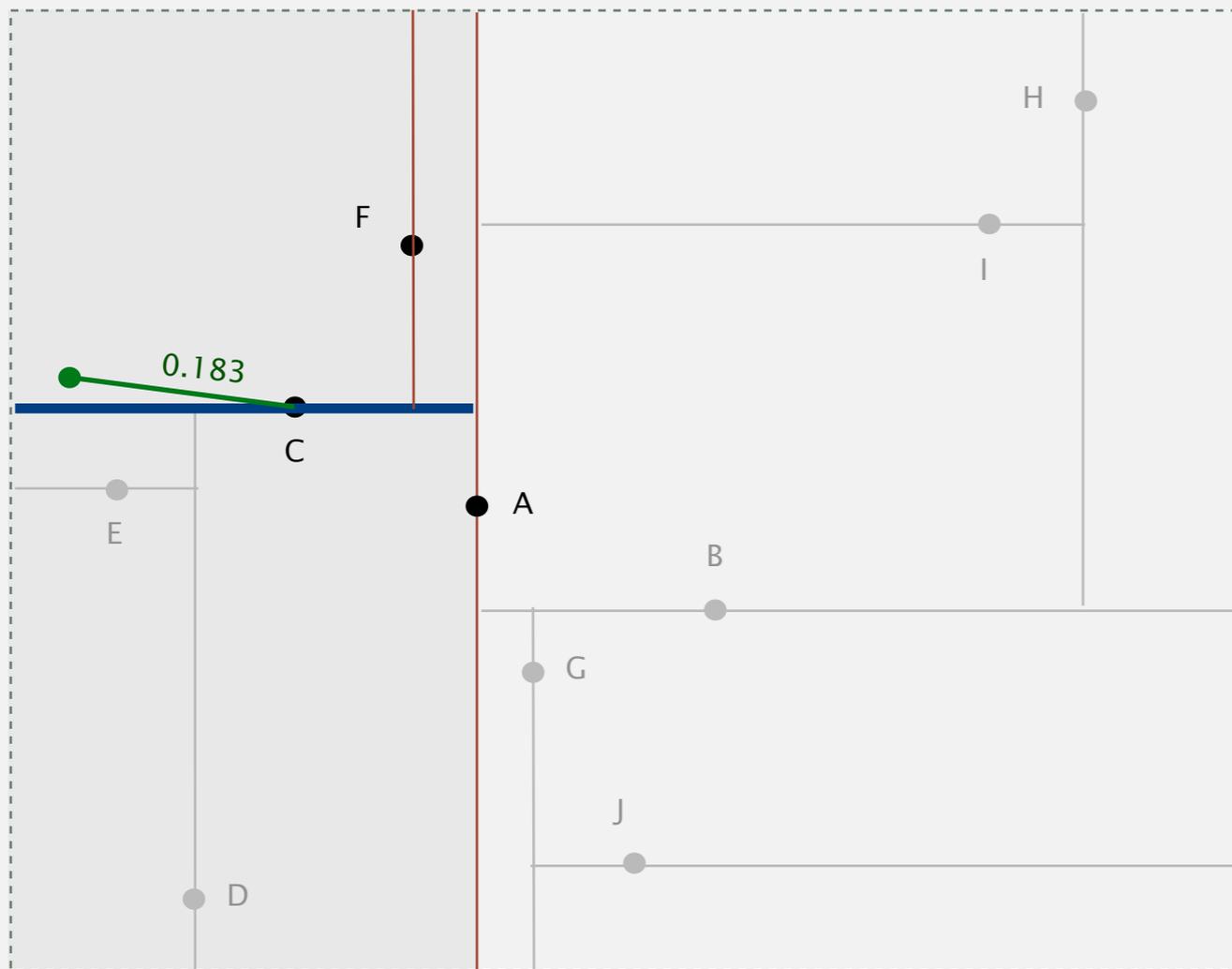
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**search right subtree
prune since nearest neighbor
can't be here**

2d tree demo: nearest neighbor

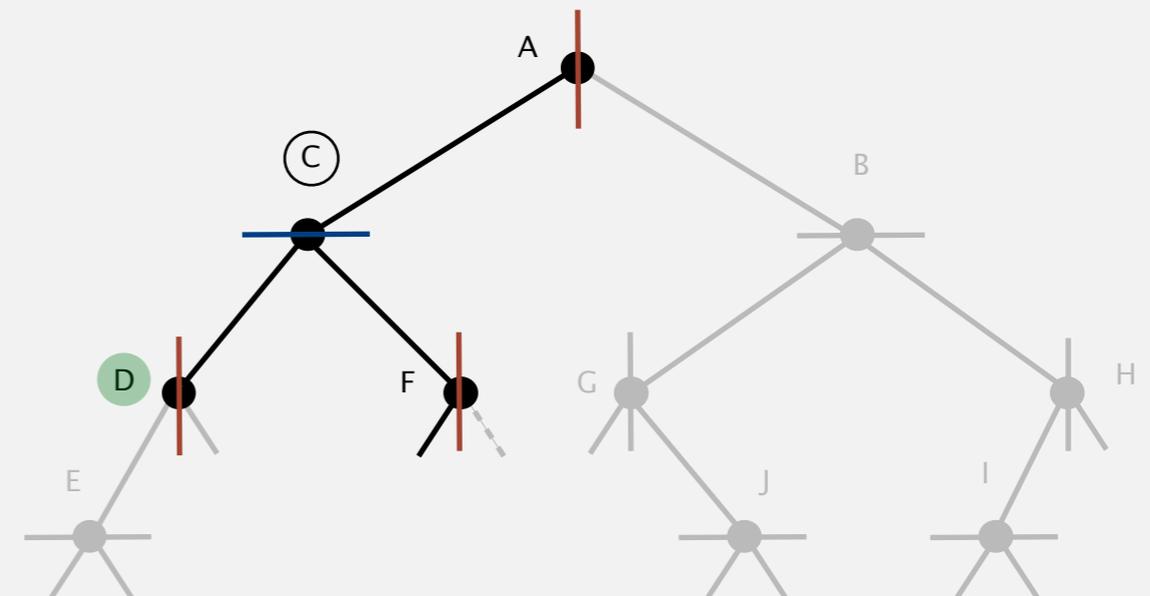
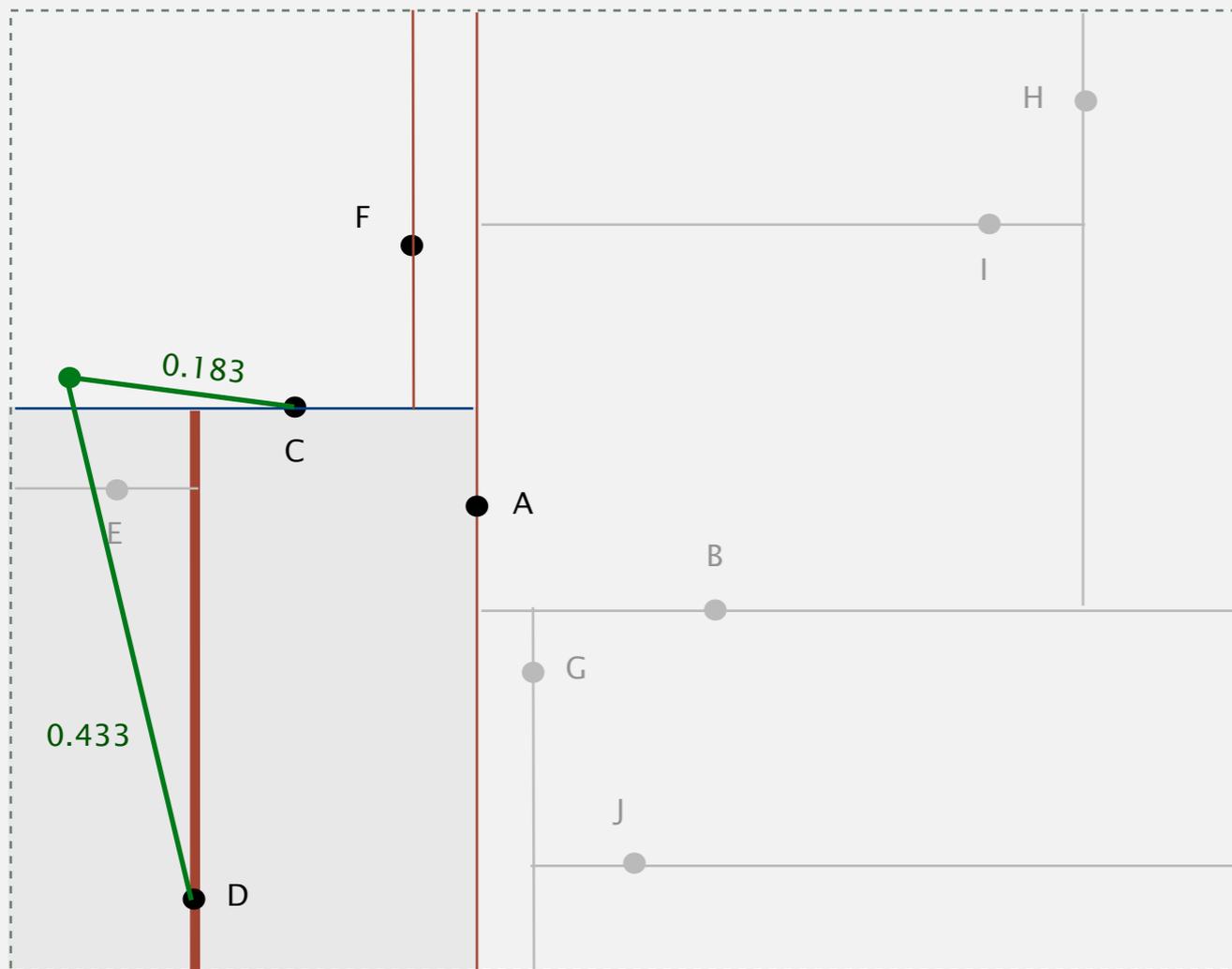
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**return from function call
search bottom subtree next**

2d tree demo: nearest neighbor

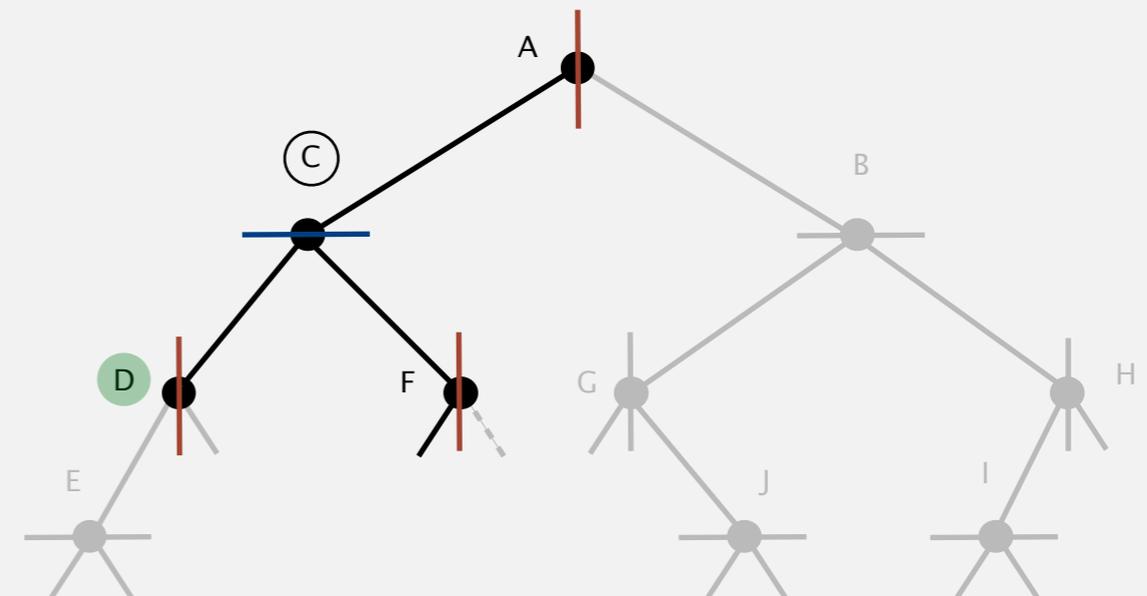
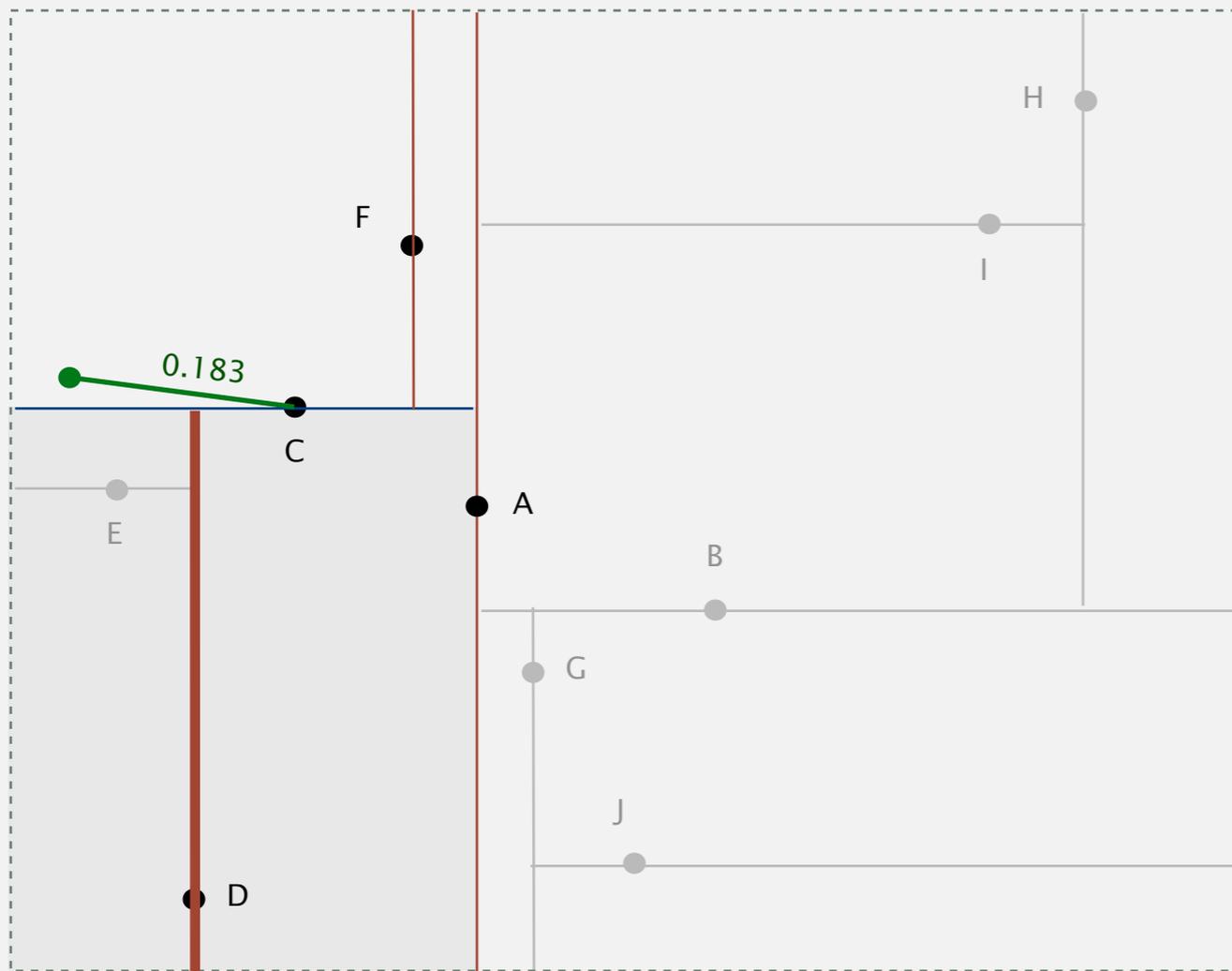
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



search bottom subtree
compute distance from query point to D

2d tree demo: nearest neighbor

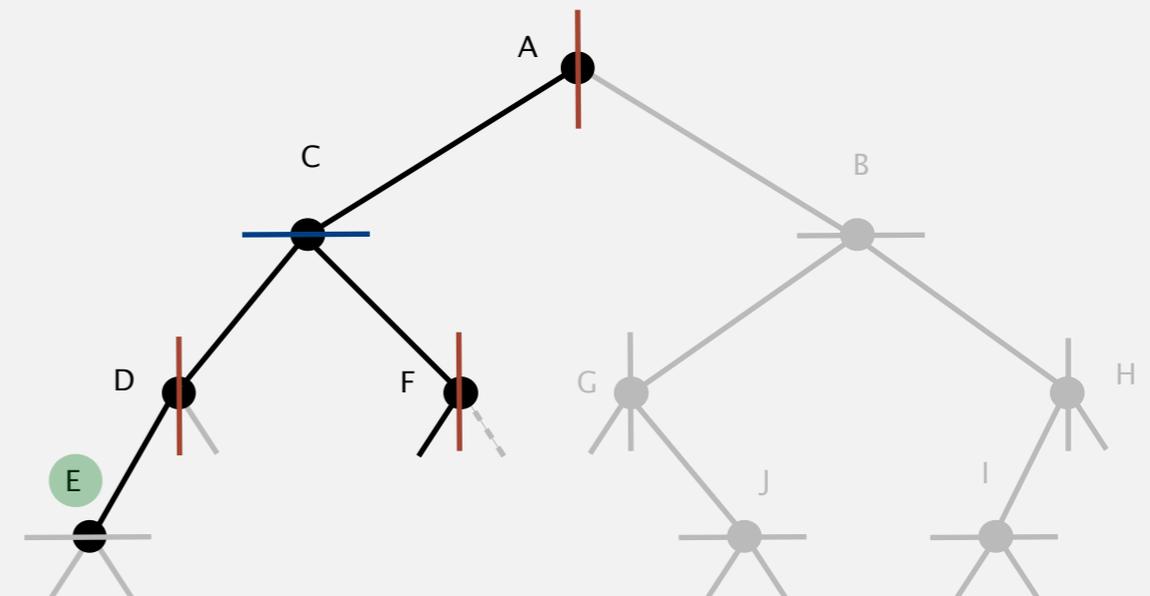
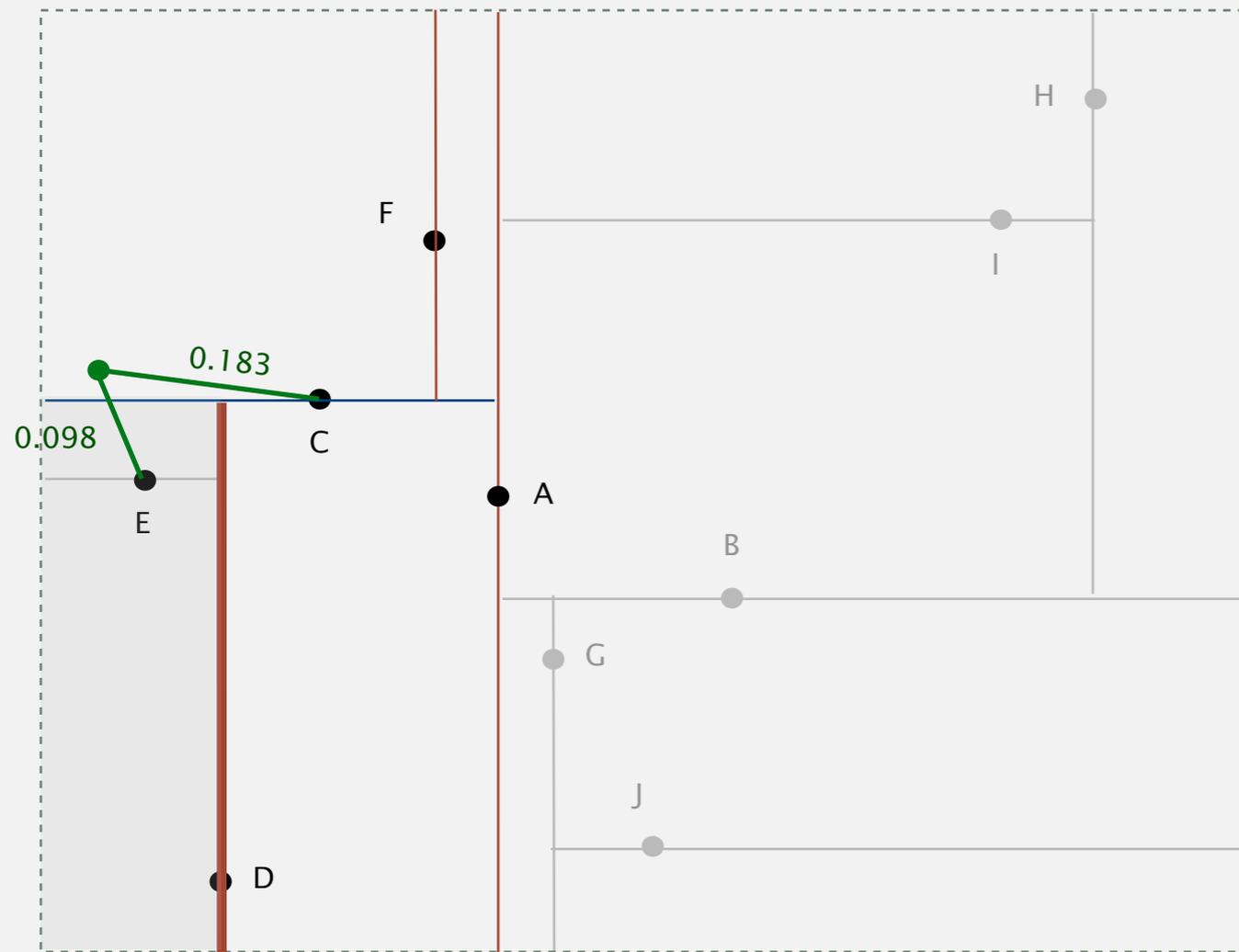
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



query point is to left of splitting line
search left subtree first

2d tree demo: nearest neighbor

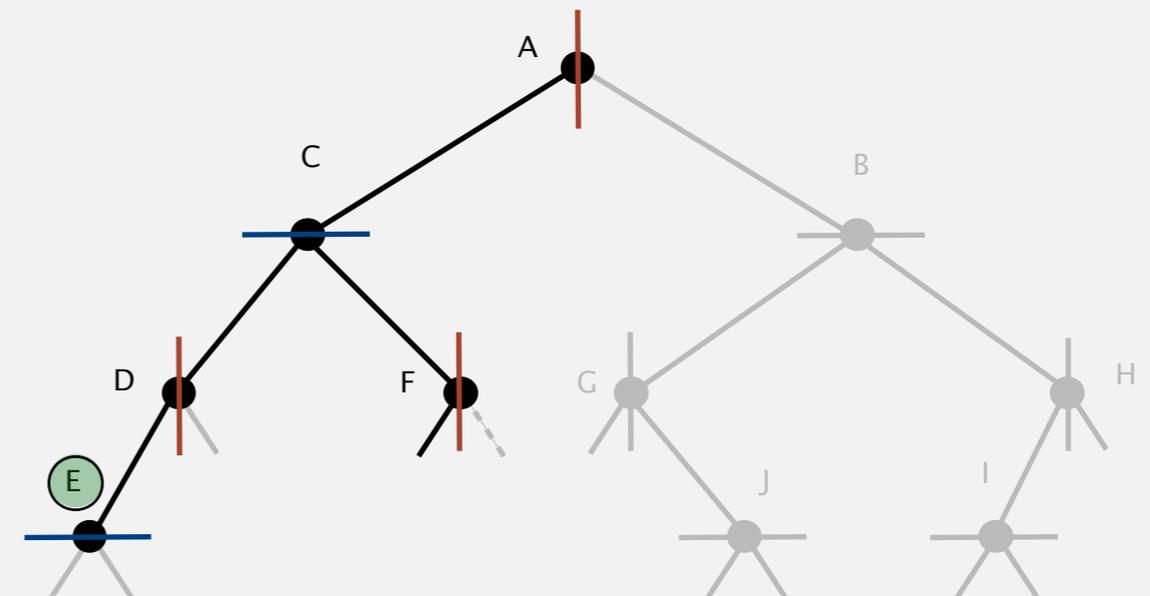
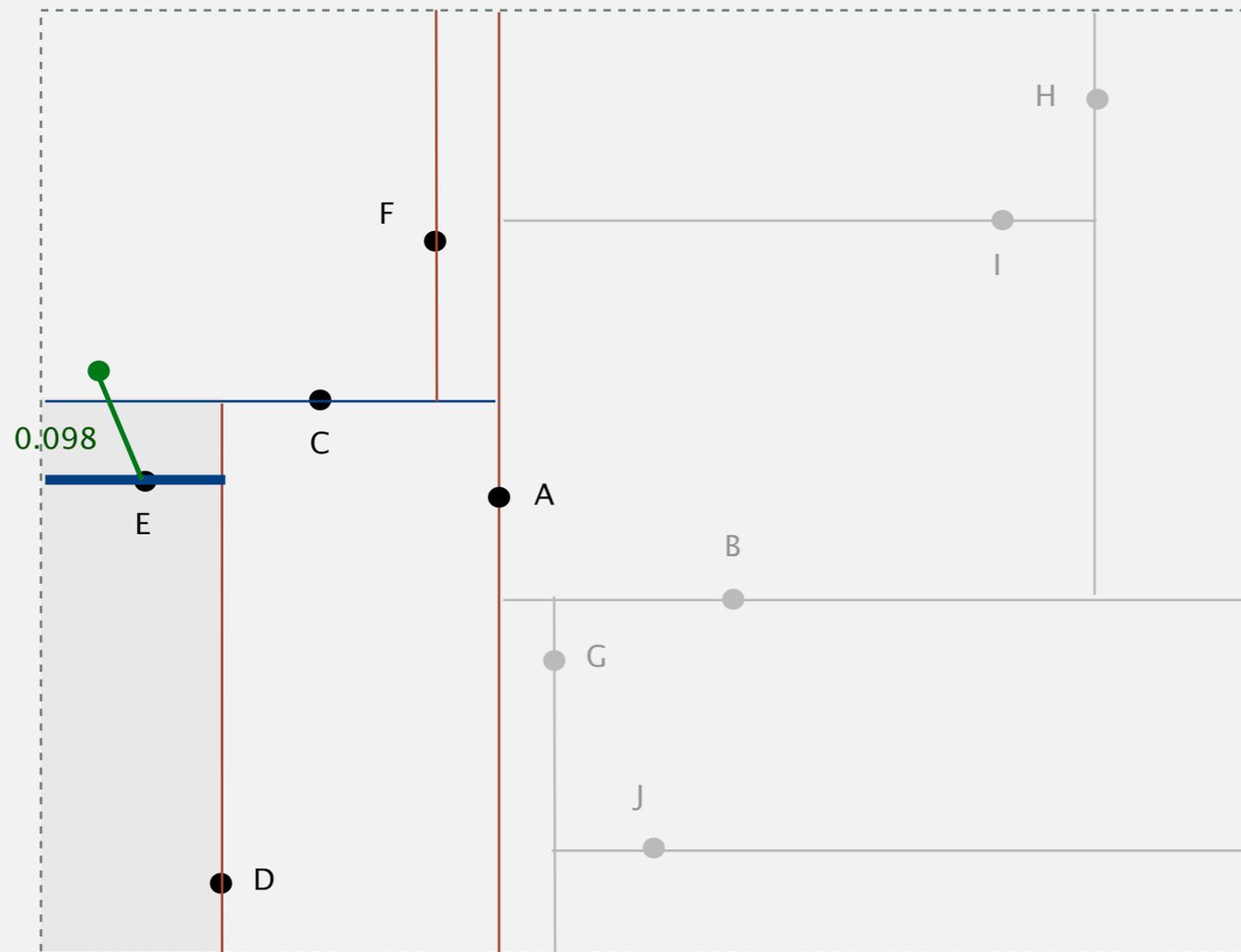
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



search left subtree
compute distance from query point to E
(update champion)

2d tree demo: nearest neighbor

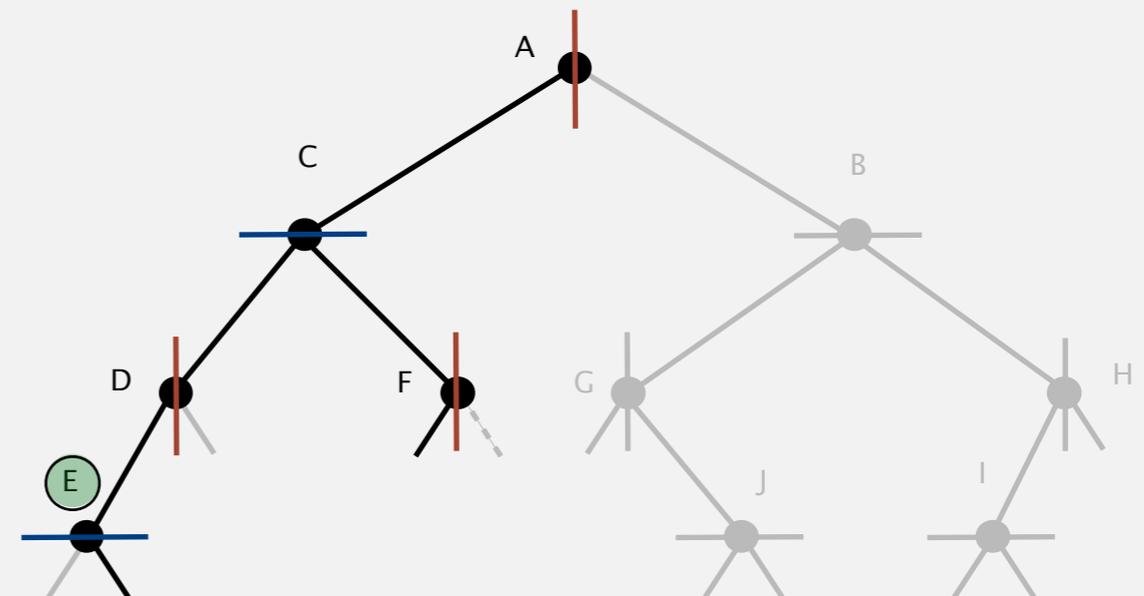
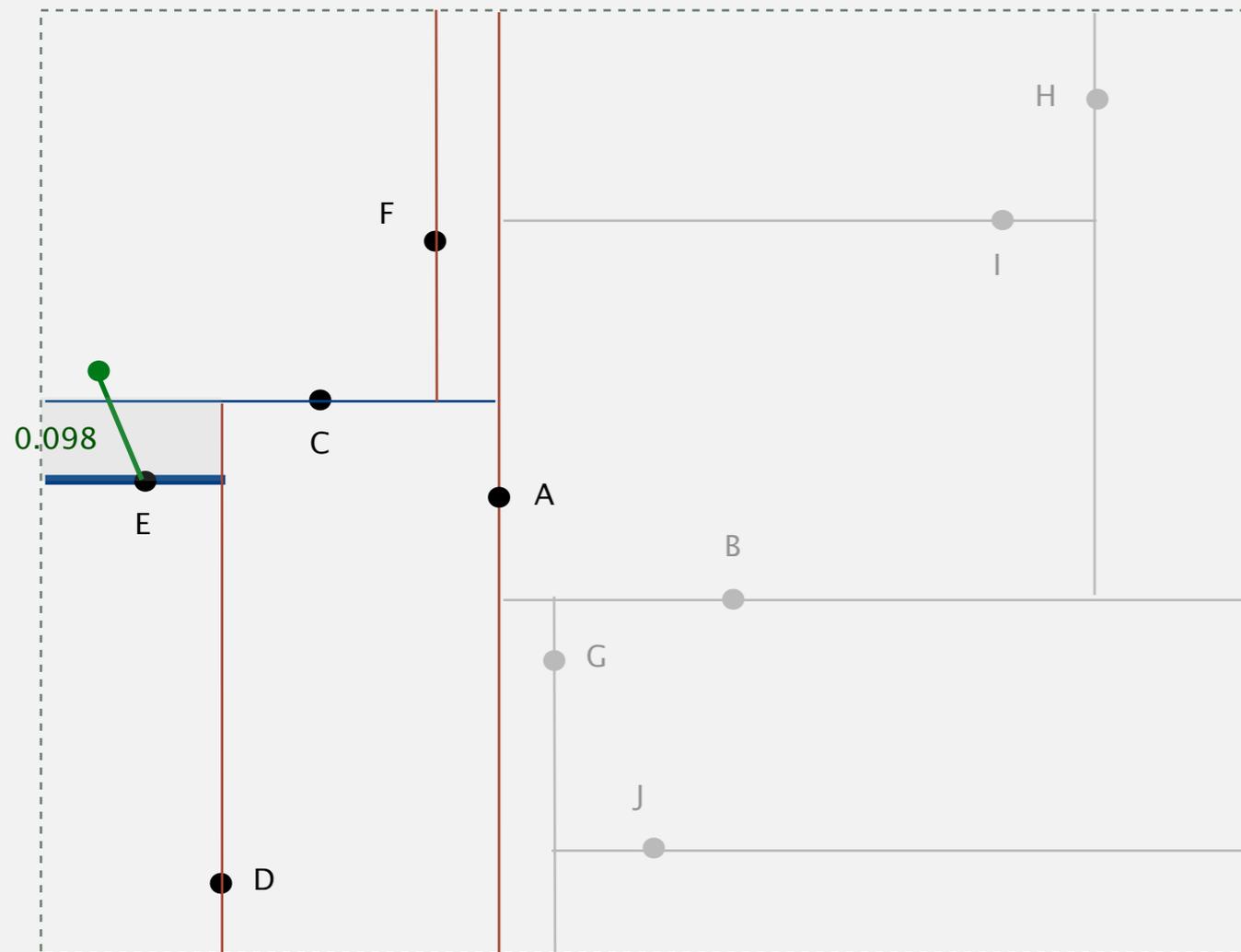
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**query point is above splitting line
search top subtree first**

2d tree demo: nearest neighbor

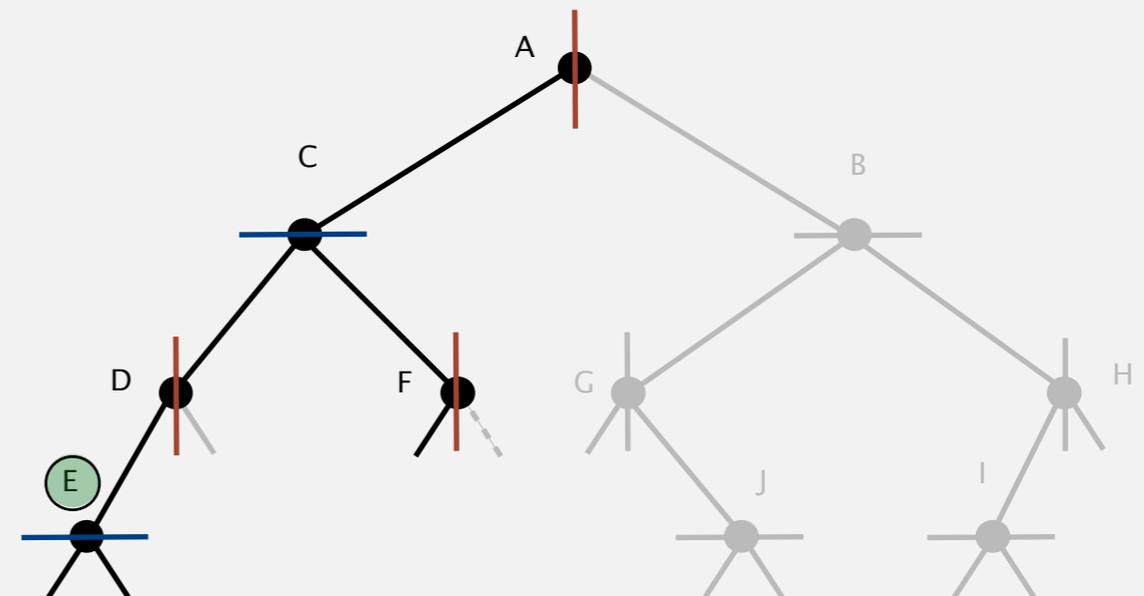
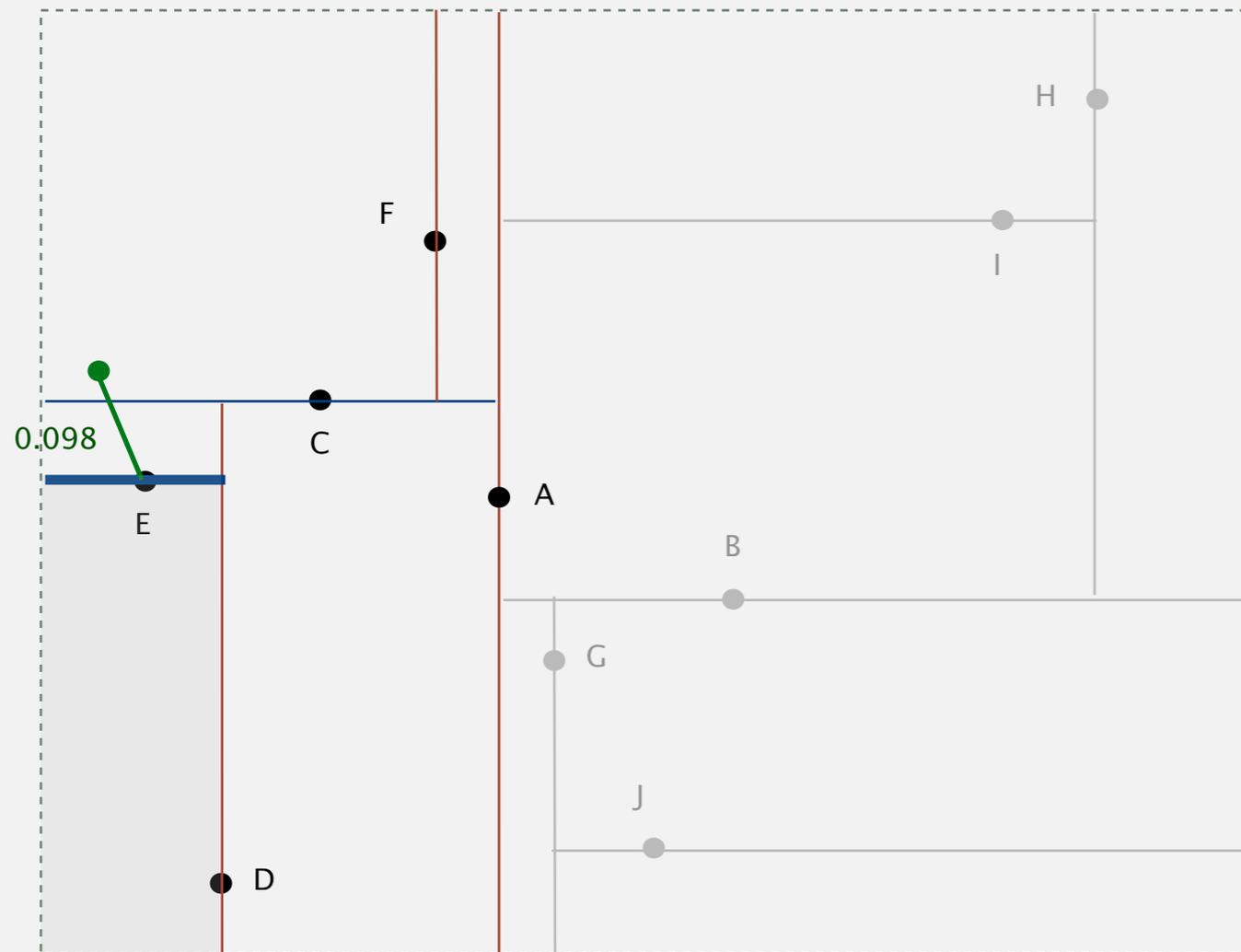
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



search top subtree
return since empty

2d tree demo: nearest neighbor

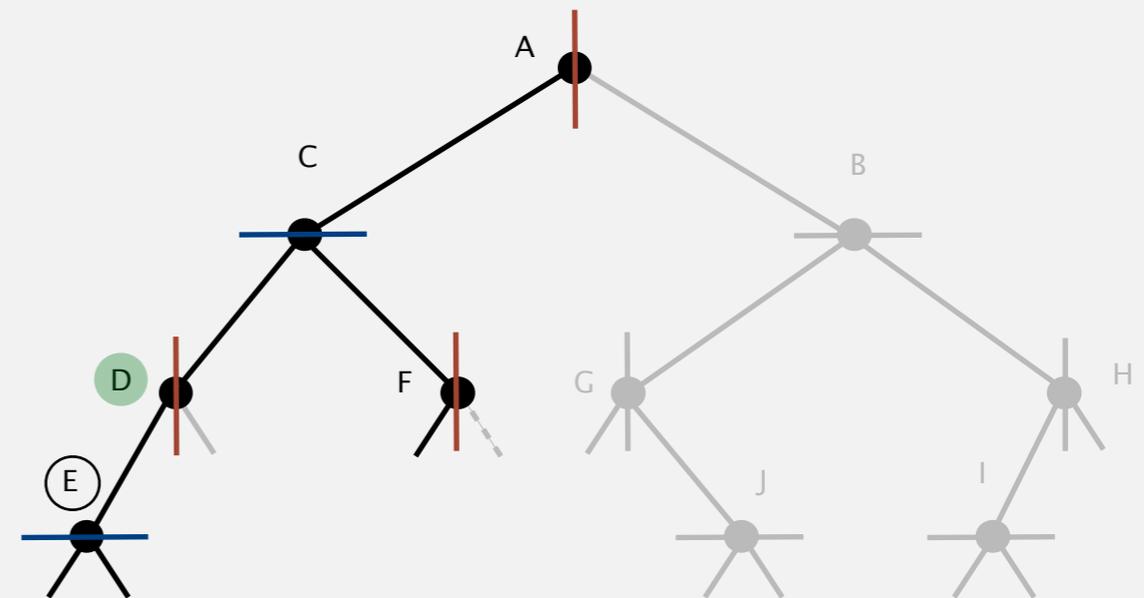
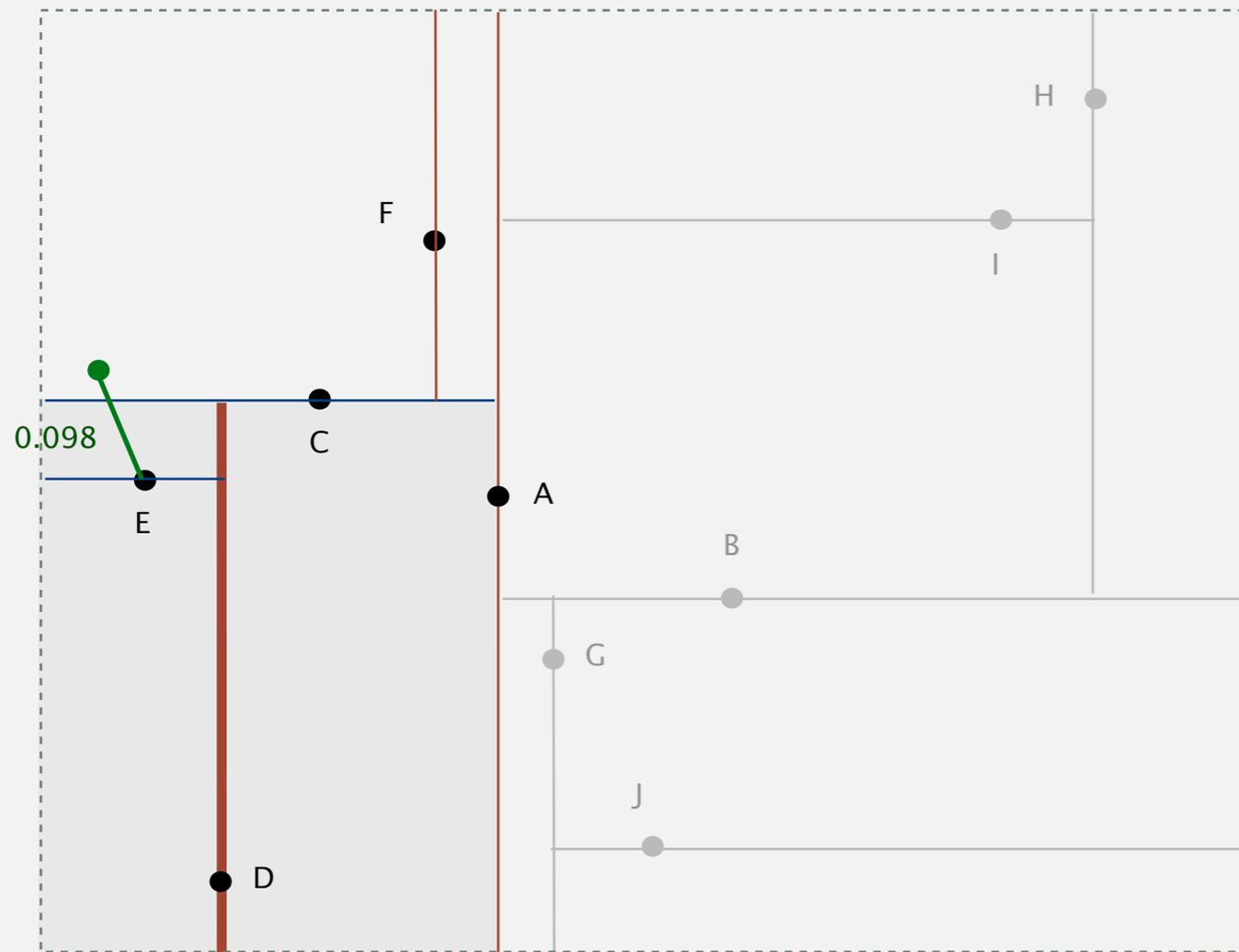
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**search bottom subtree
return since empty**

2d tree demo: nearest neighbor

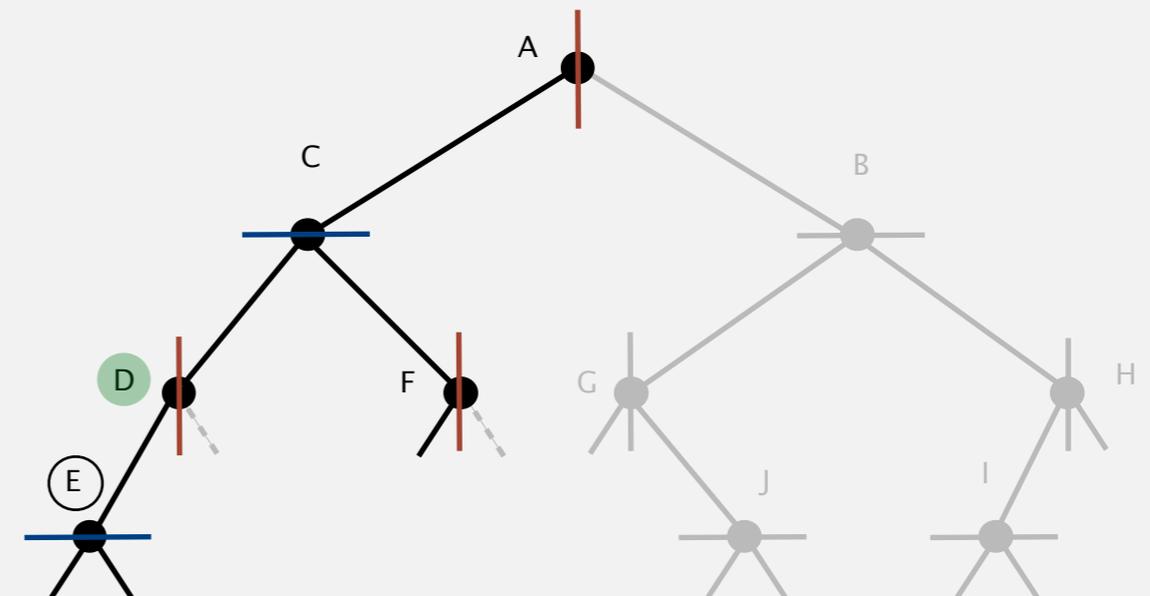
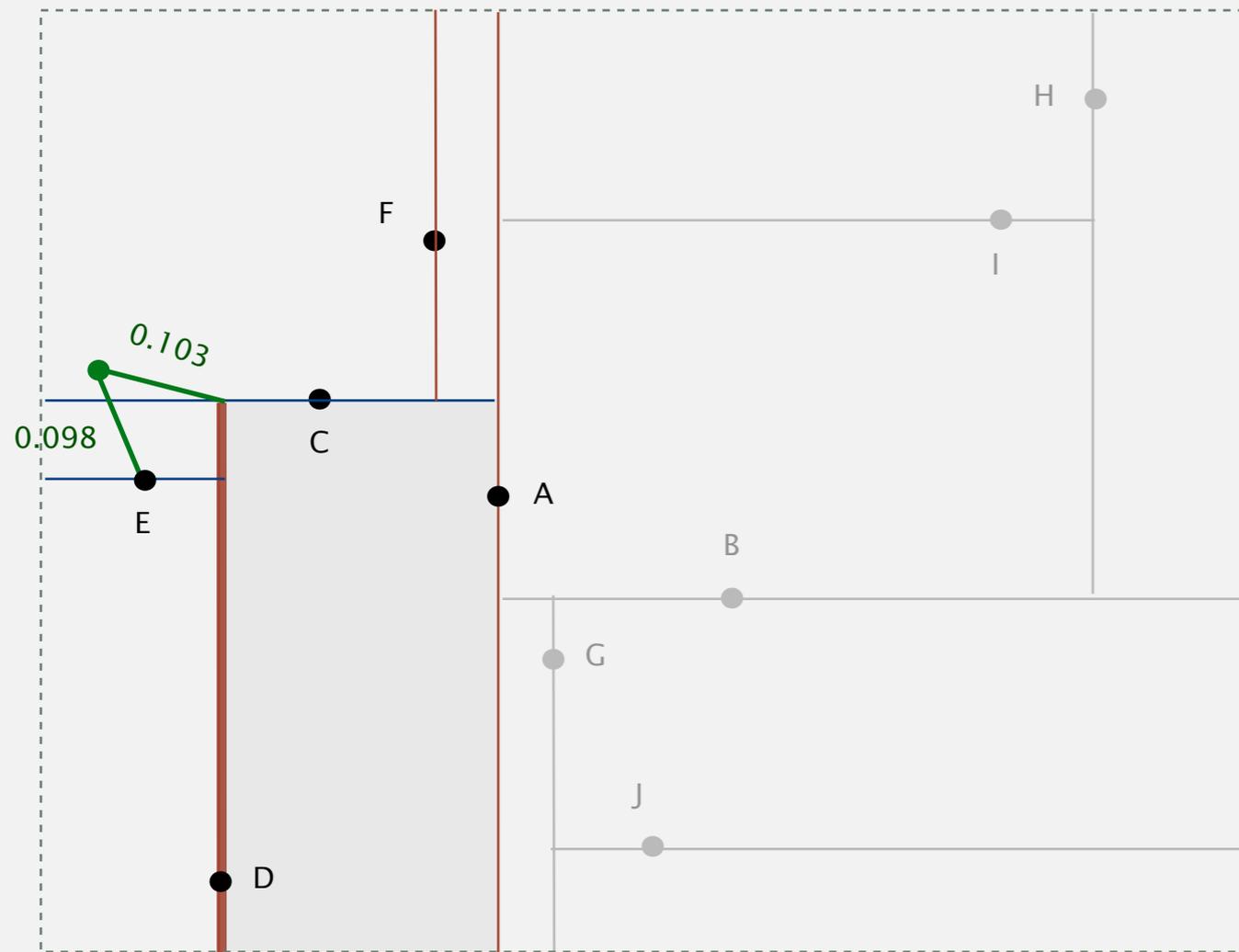
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



return from function call
search right subtree next

2d tree demo: nearest neighbor

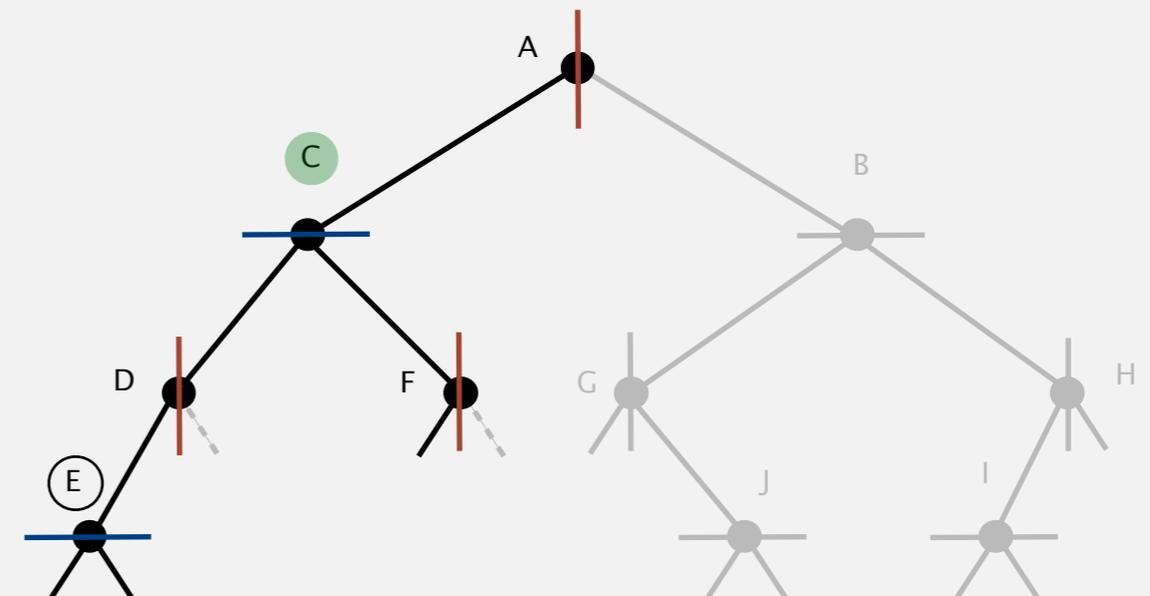
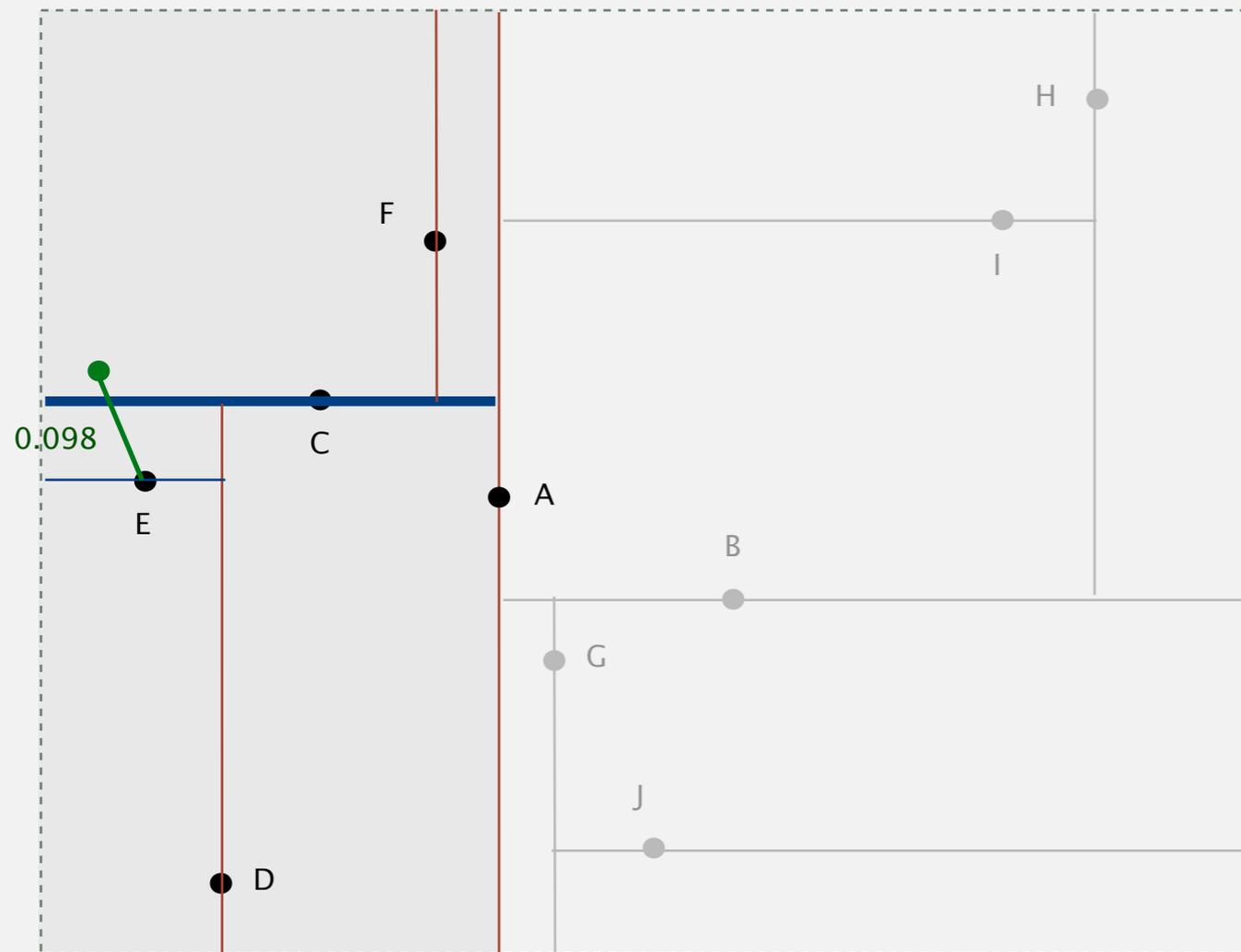
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**search right subtree
prune since nearest neighbor
can't be here**

2d tree demo: nearest neighbor

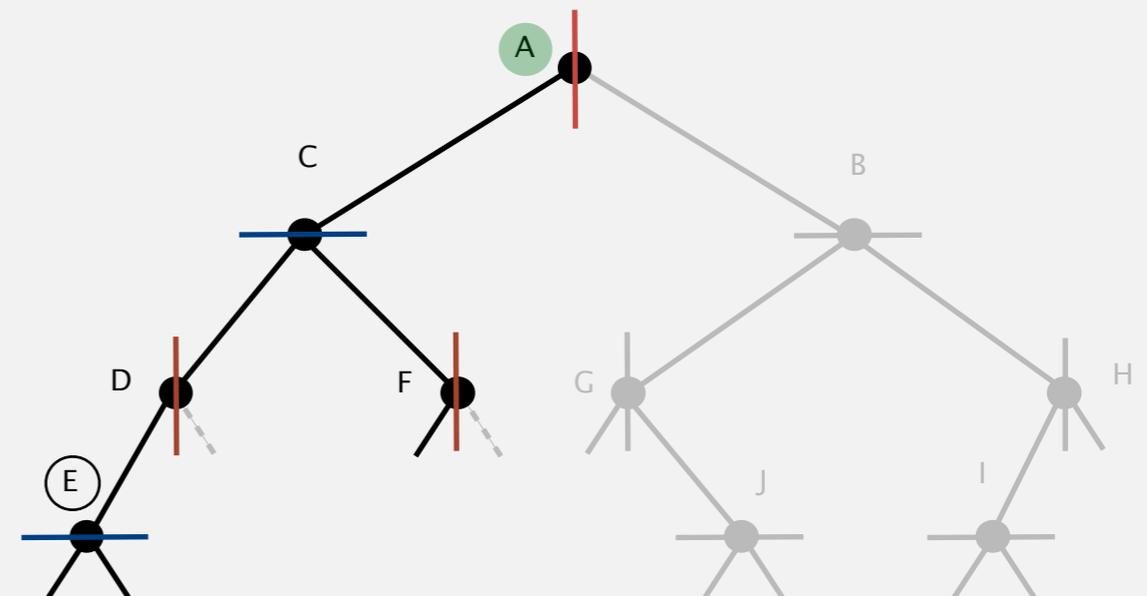
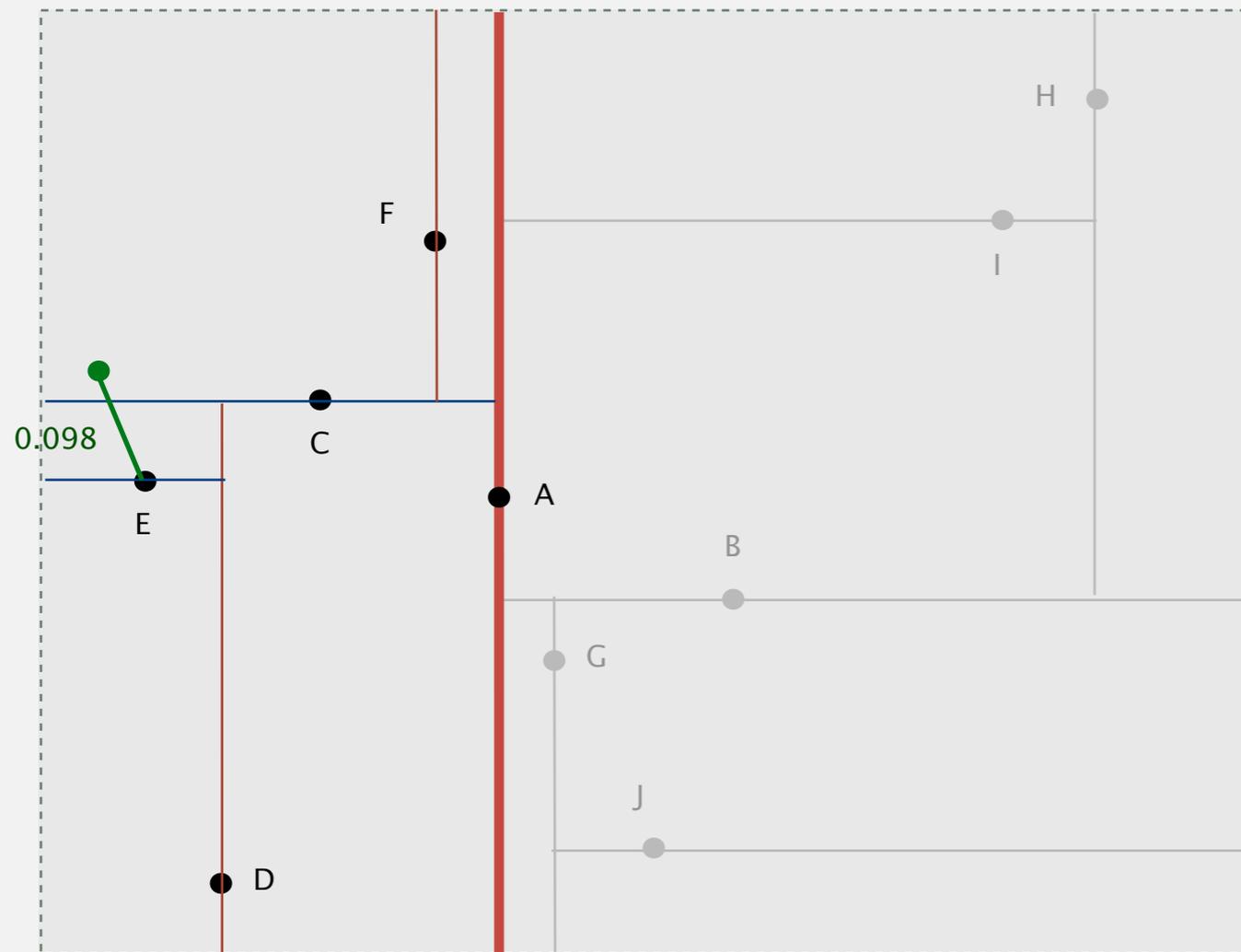
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



return from function call

2d tree demo: nearest neighbor

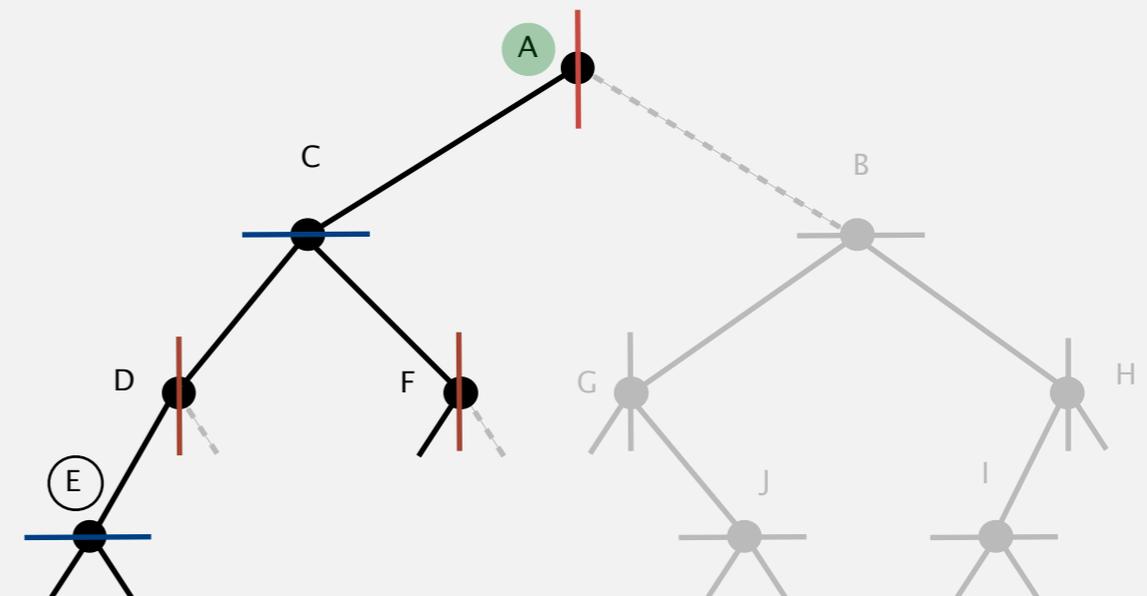
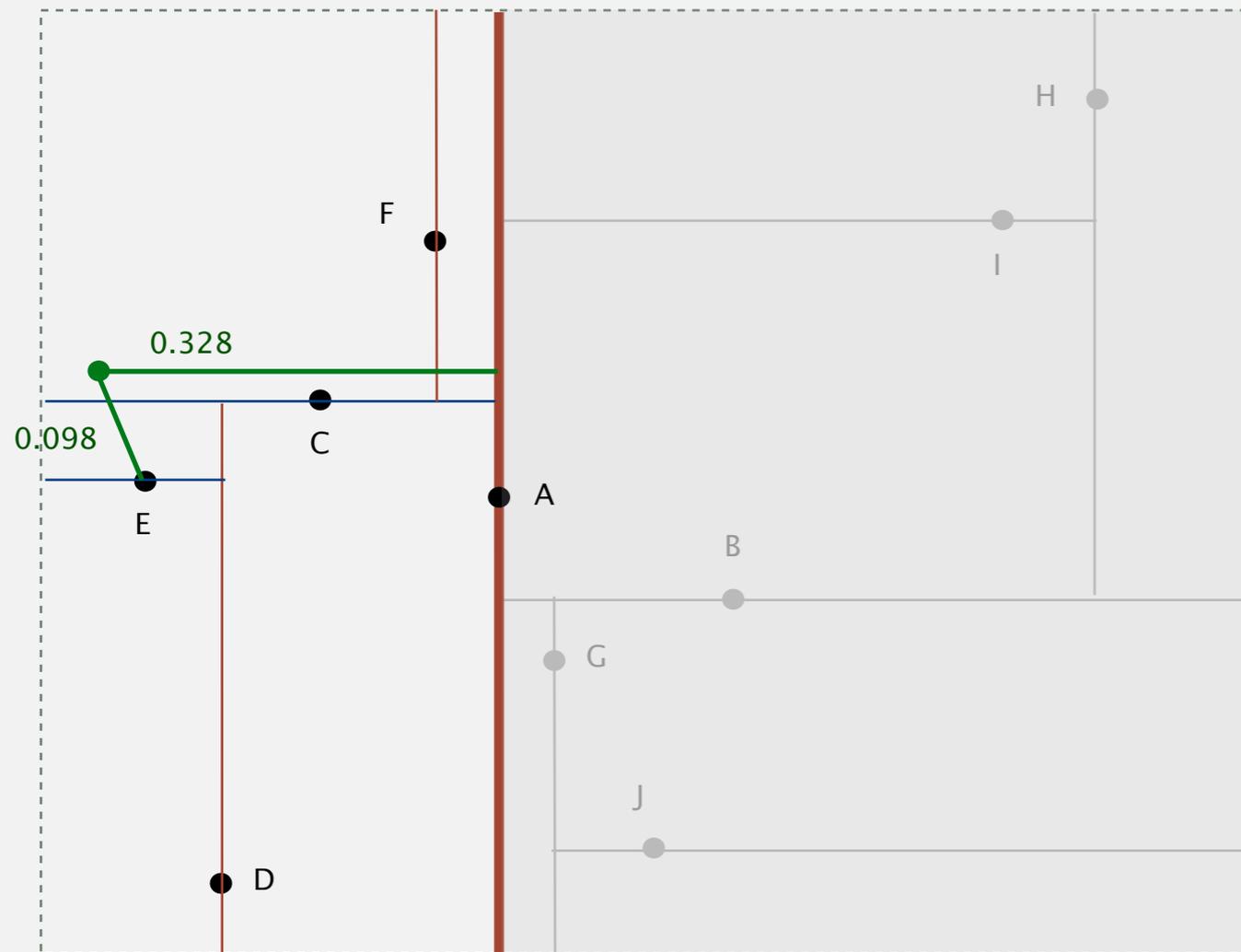
- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**return from function call
search right subtree next**

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom (if it could contain a closer point).
- Recursively search right/top (if it could contain a closer point).
- Organize method so that it begins by searching for query point.



**search right subtree
prune since nearest neighbor
can't be here**

