https://algs4.cs.princeton.edu

2D TREE DEMO

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

Last updated on 3/4/20 3:31 PM

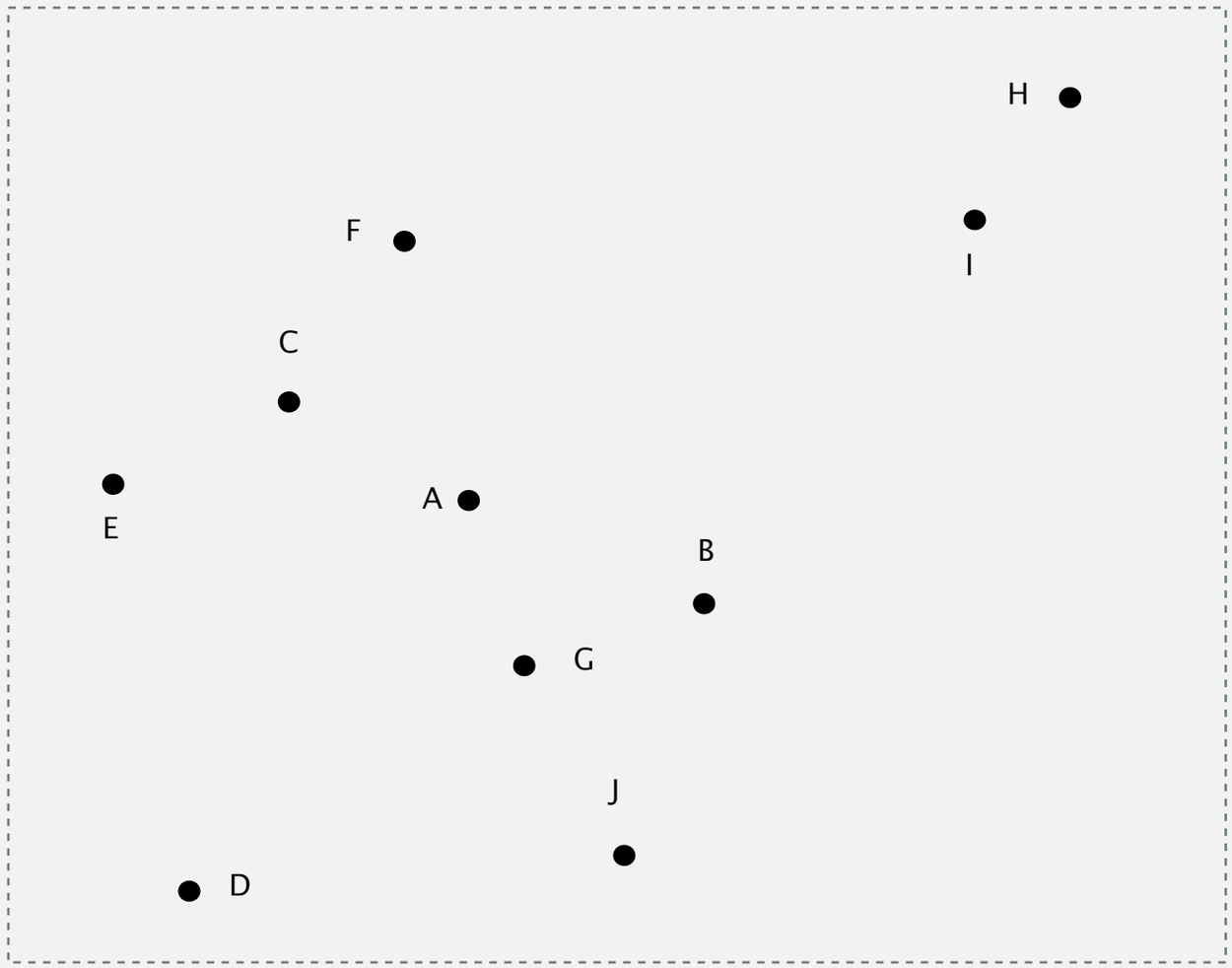
https://algs4.cs.princeton.edu

2d Tree Demo

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

2d tree demo

Points to insert.



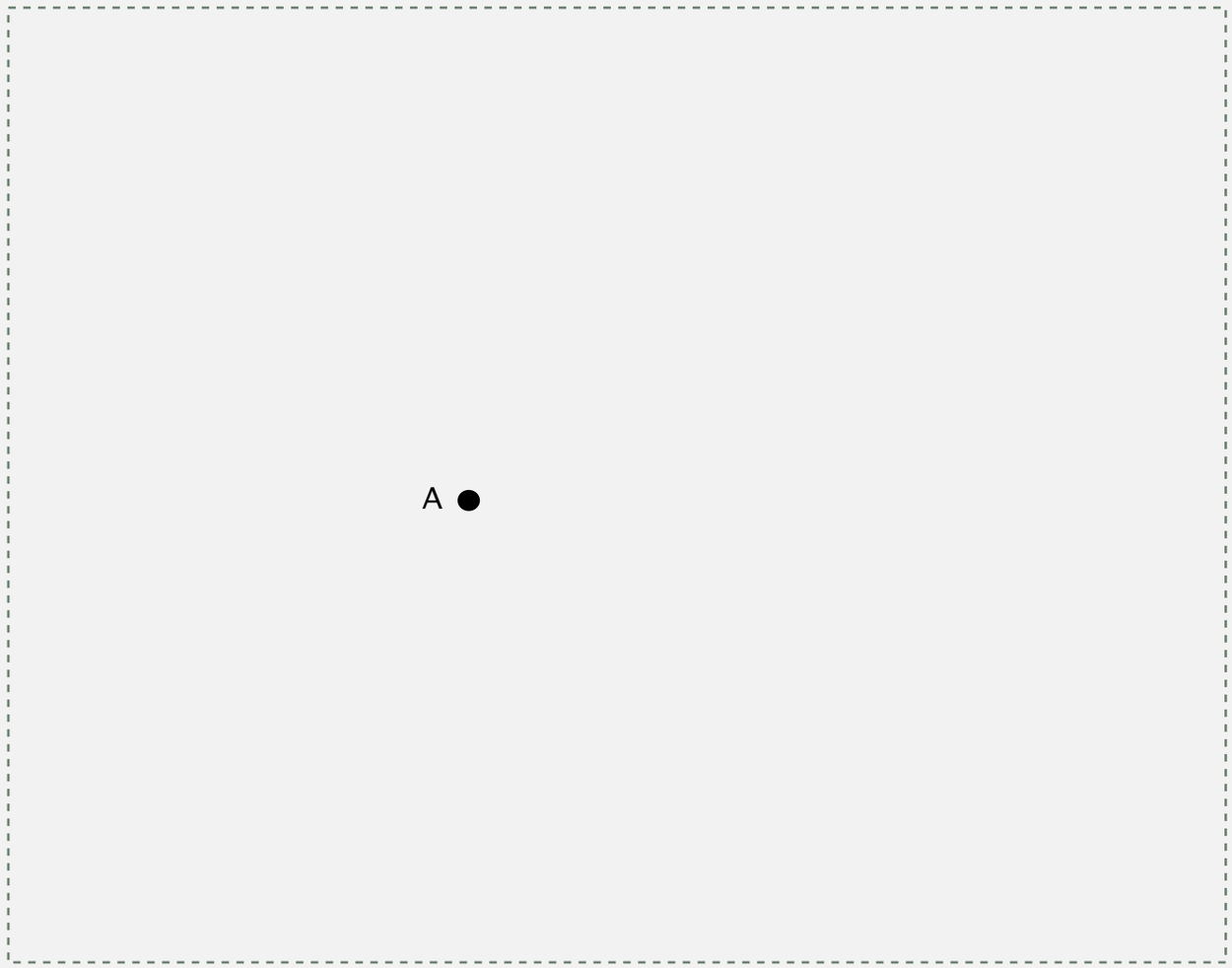

```
% more input10.txt
0.372 0.497
0.564 0.413
0.226 0.577
0.144 0.179
0.083 0.510
0.320 0.708
0.417 0.362
0.862 0.825
0.785 0.725
0.499 0.208
```

2d tree demo: insertion

Recursively partition plane into two halfplanes.

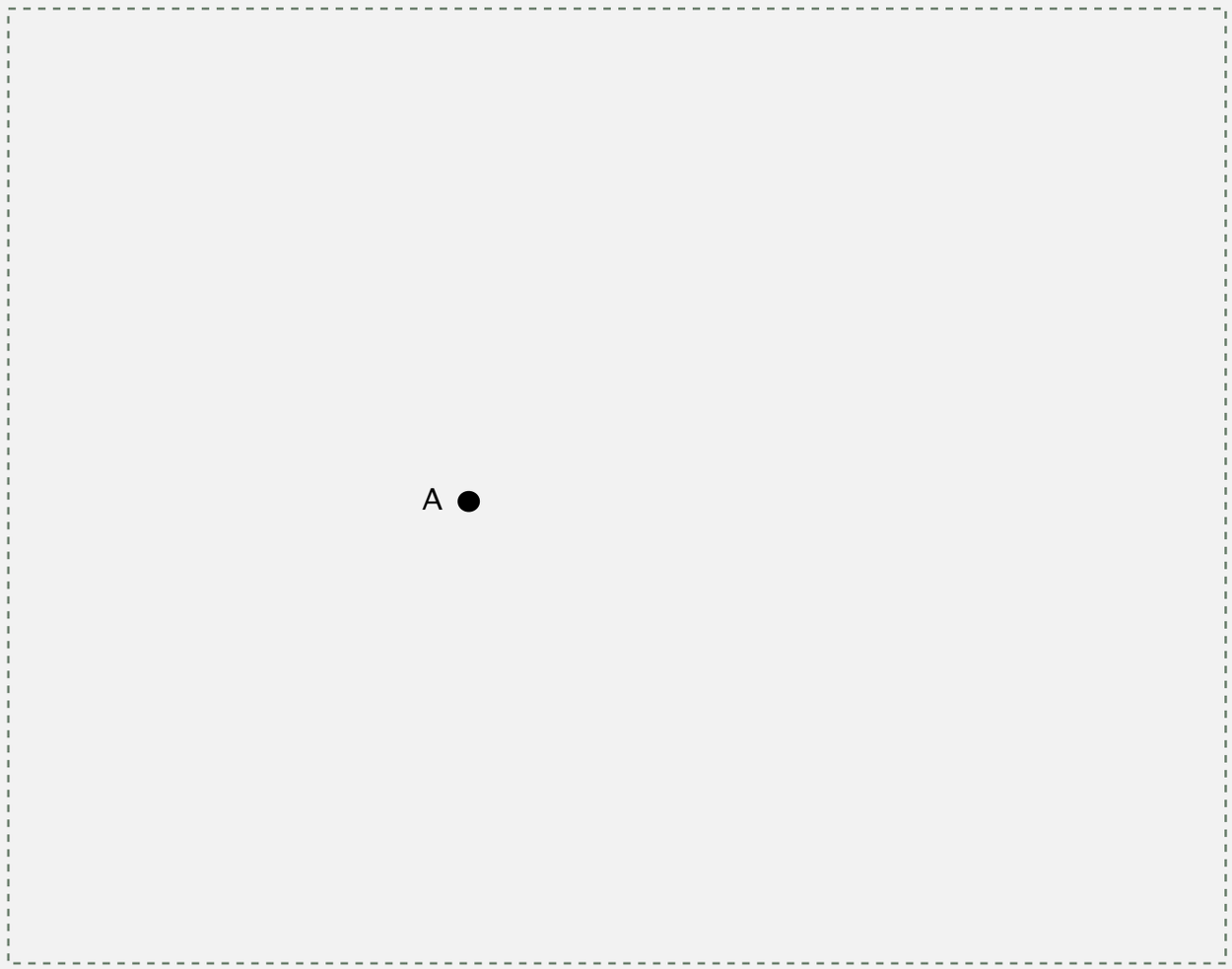
2d tree demo: insertion

Recursively partition plane into two halfplanes.



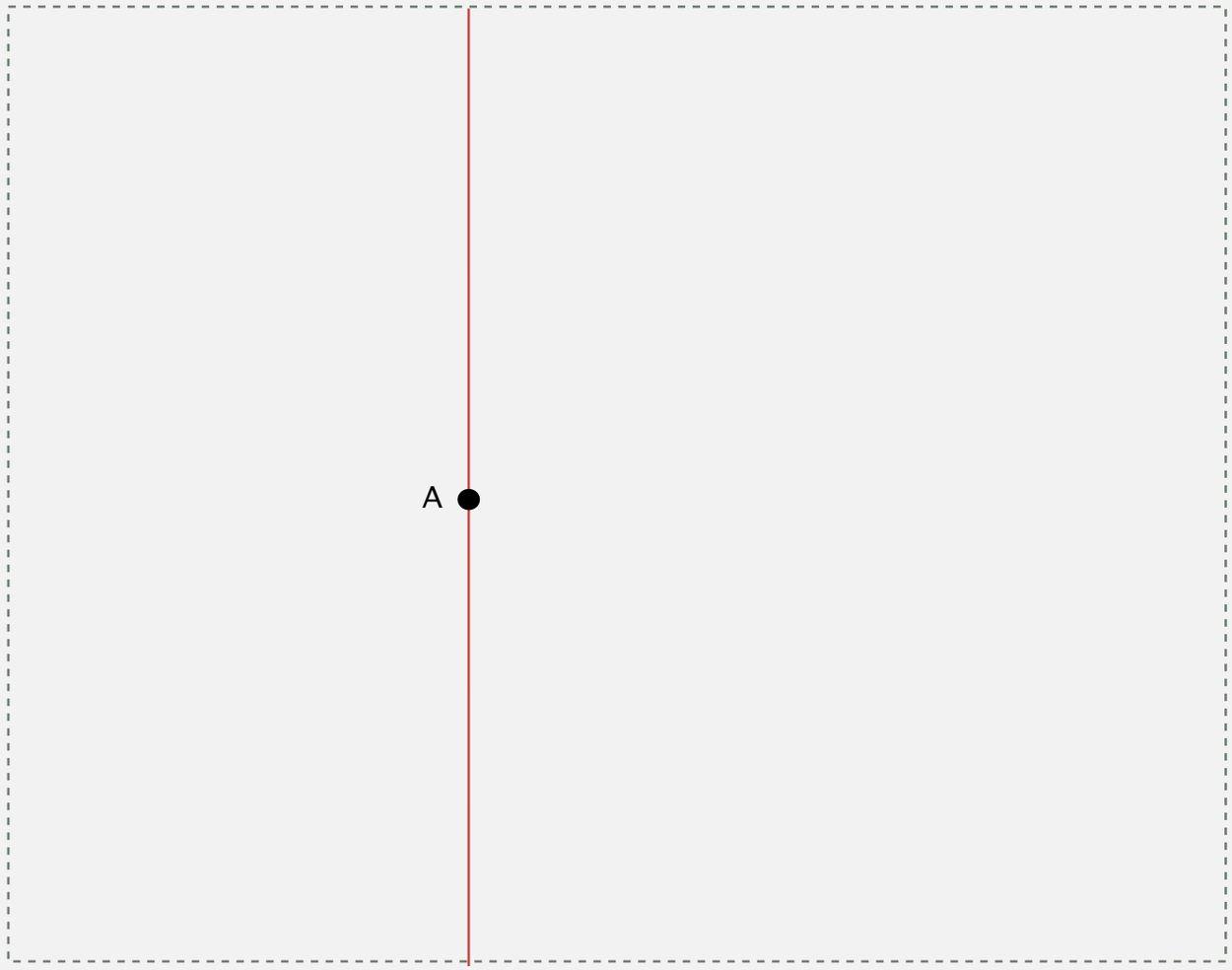
2d tree demo: insertion

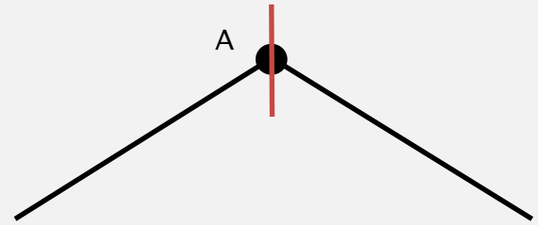
Recursively partition plane into two halfplanes.



2d tree demo: insertion

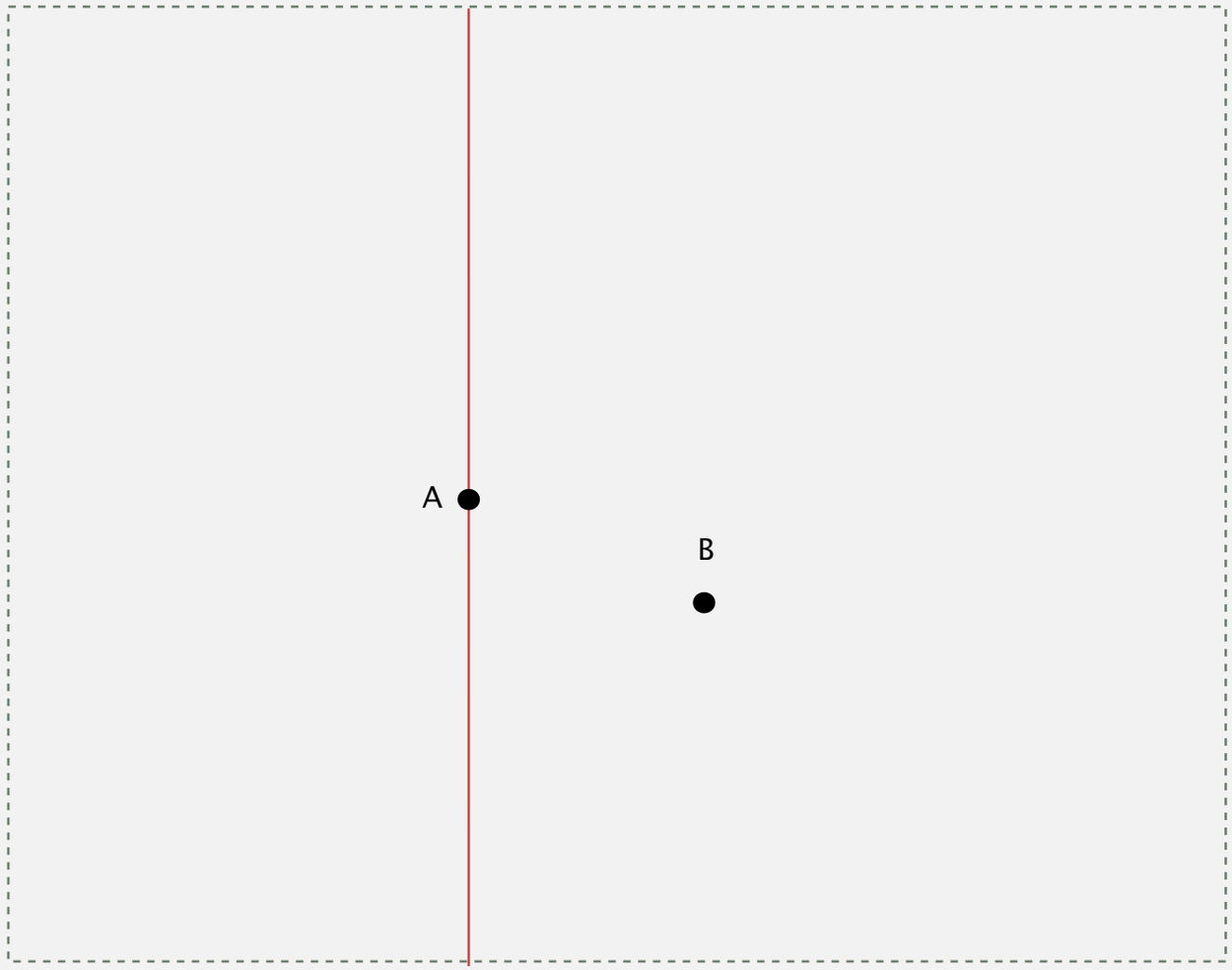
Recursively partition plane into two halfplanes.

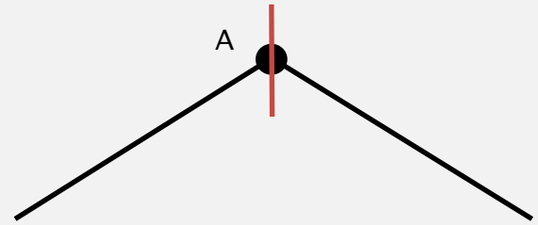




2d tree demo: insertion

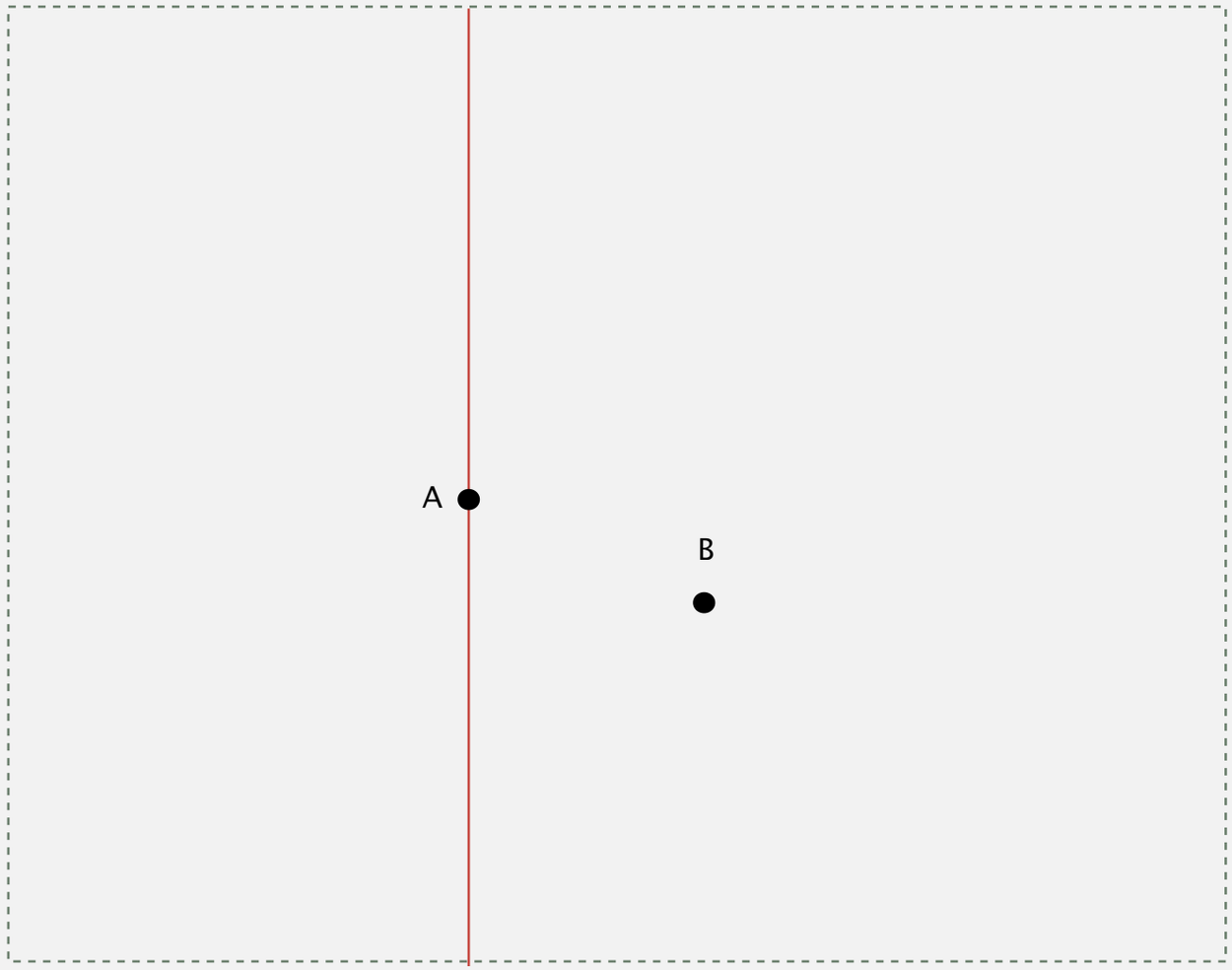
Recursively partition plane into two halfplanes.

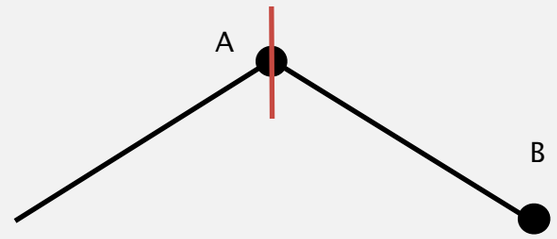




2d tree demo: insertion

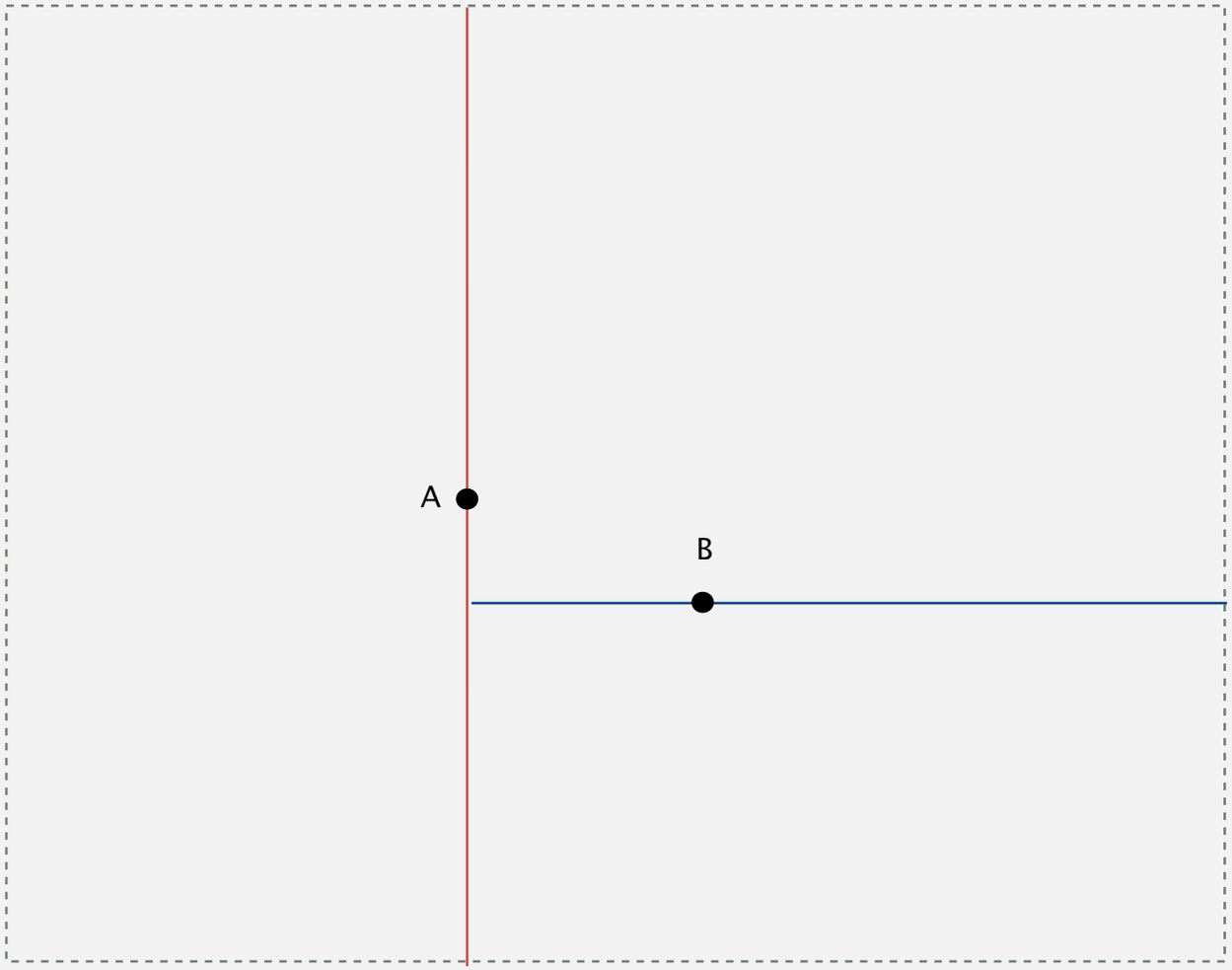
Recursively partition plane into two halfplanes.

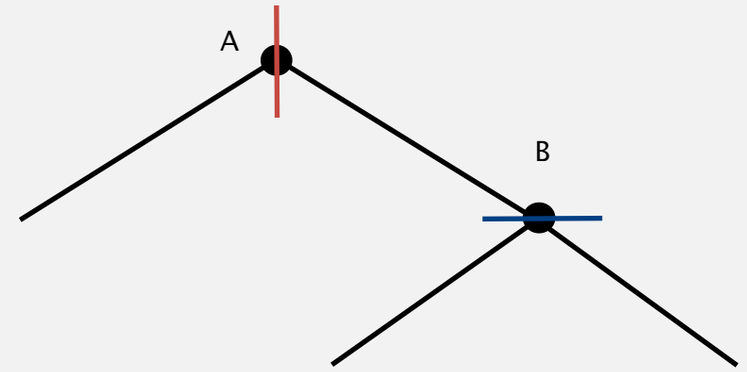




2d tree demo: insertion

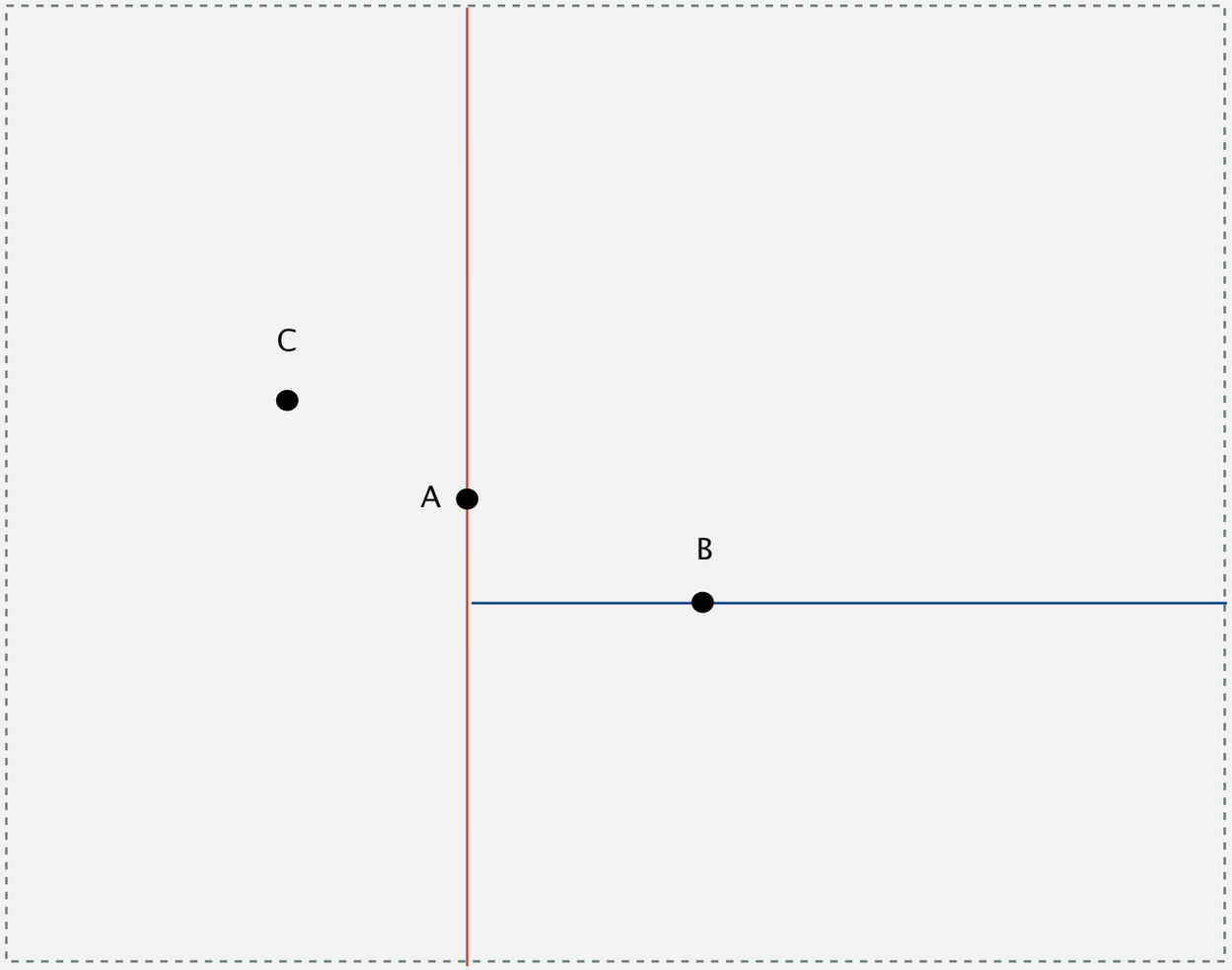
Recursively partition plane into two halfplanes.

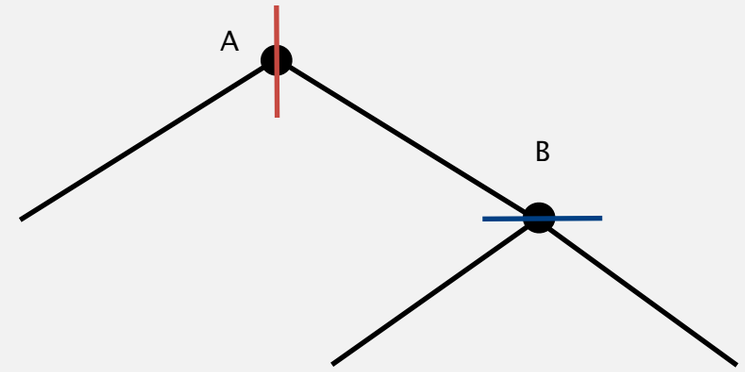




2d tree demo: insertion

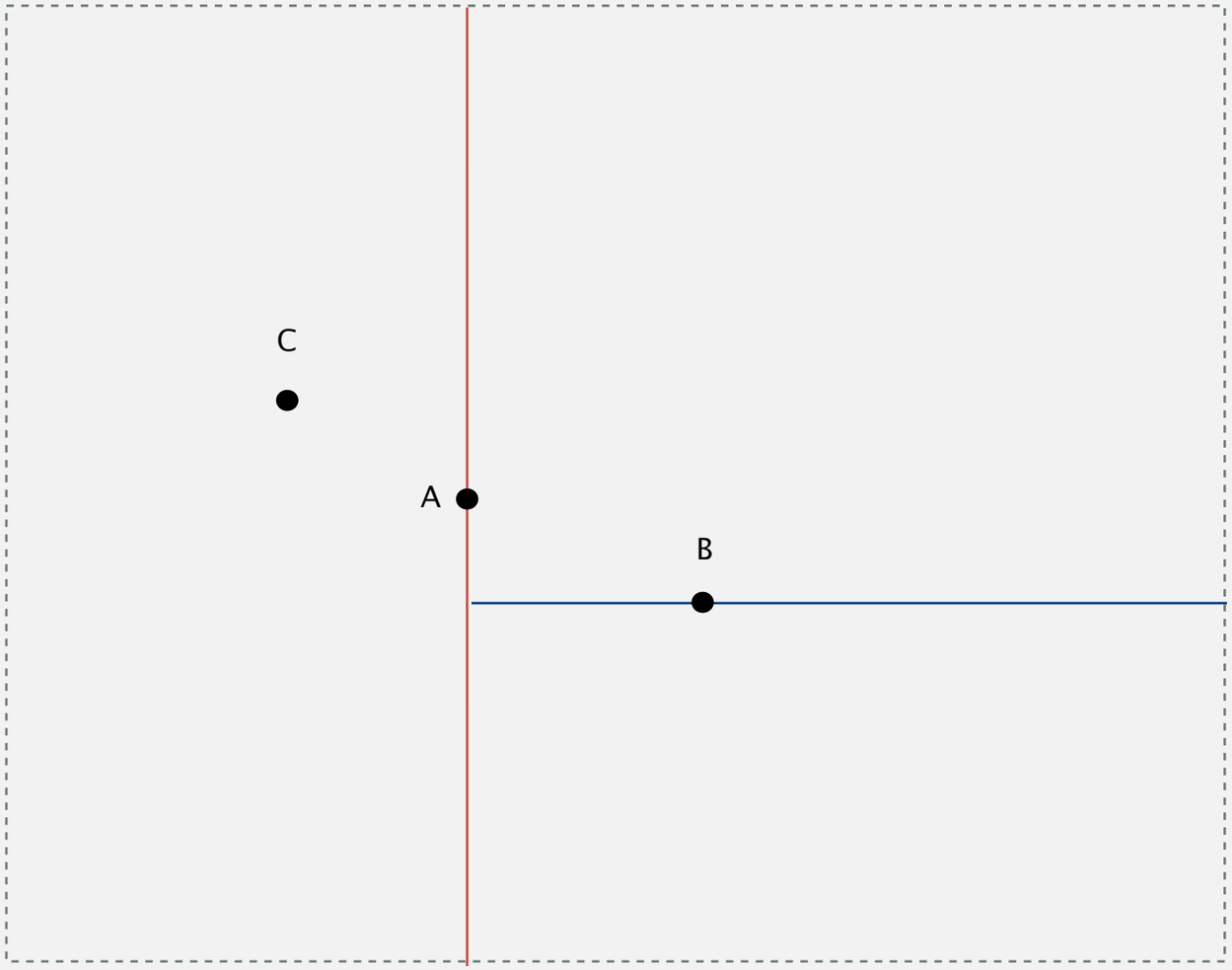
Recursively partition plane into two halfplanes.

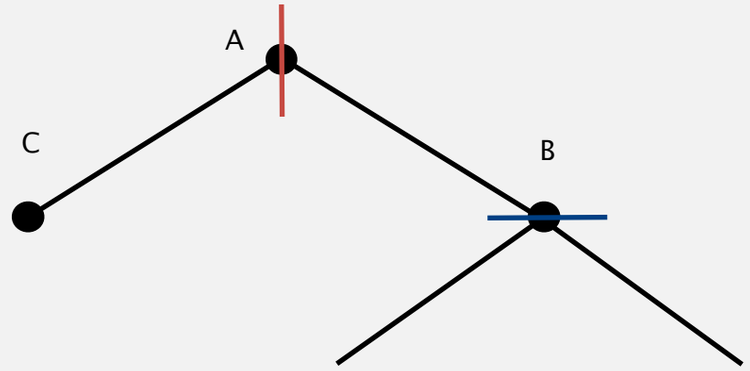




2d tree demo: insertion

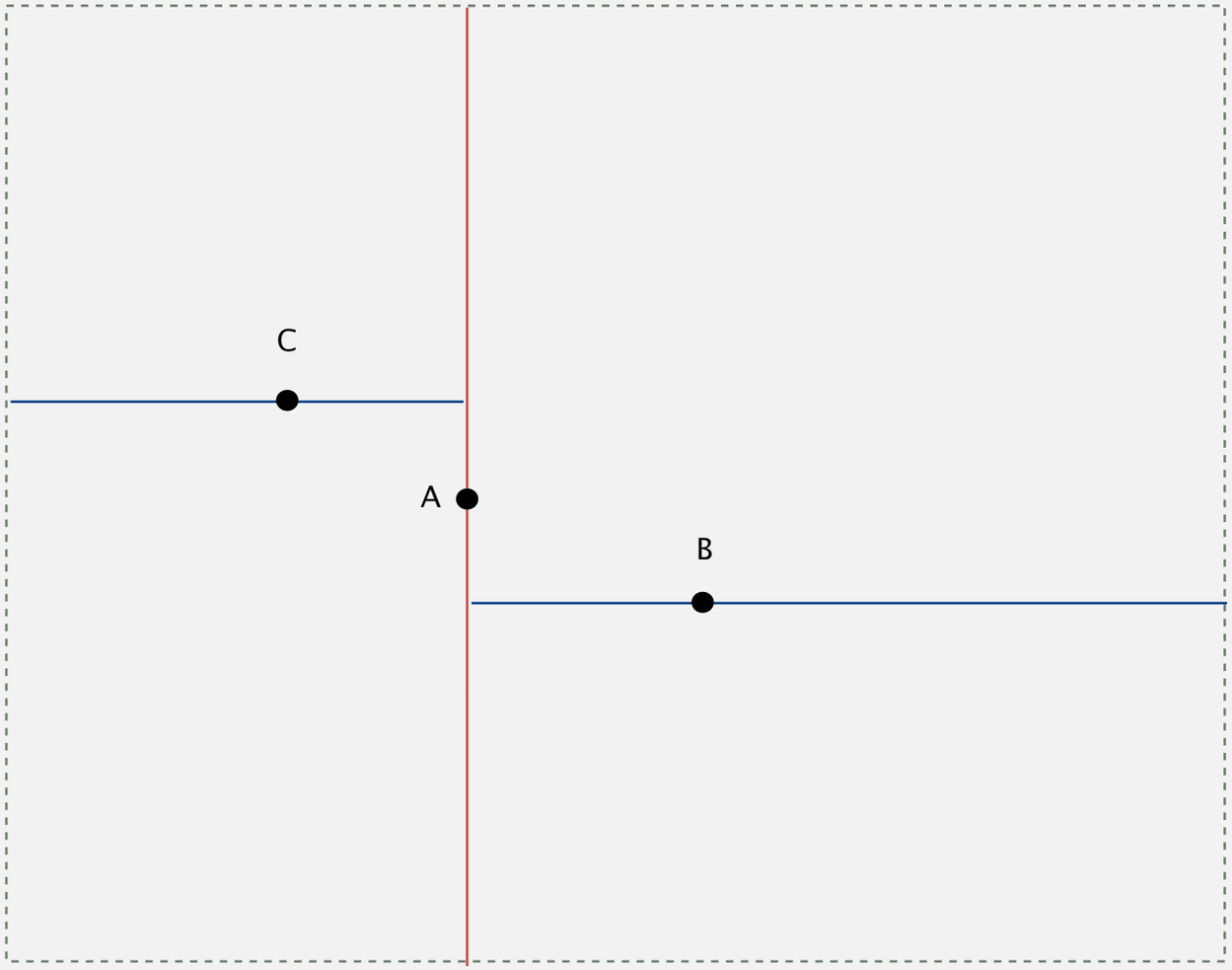
Recursively partition plane into two halfplanes.

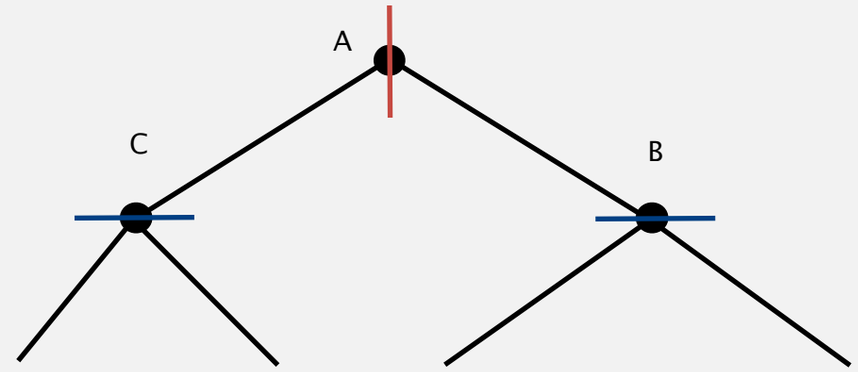




2d tree demo: insertion

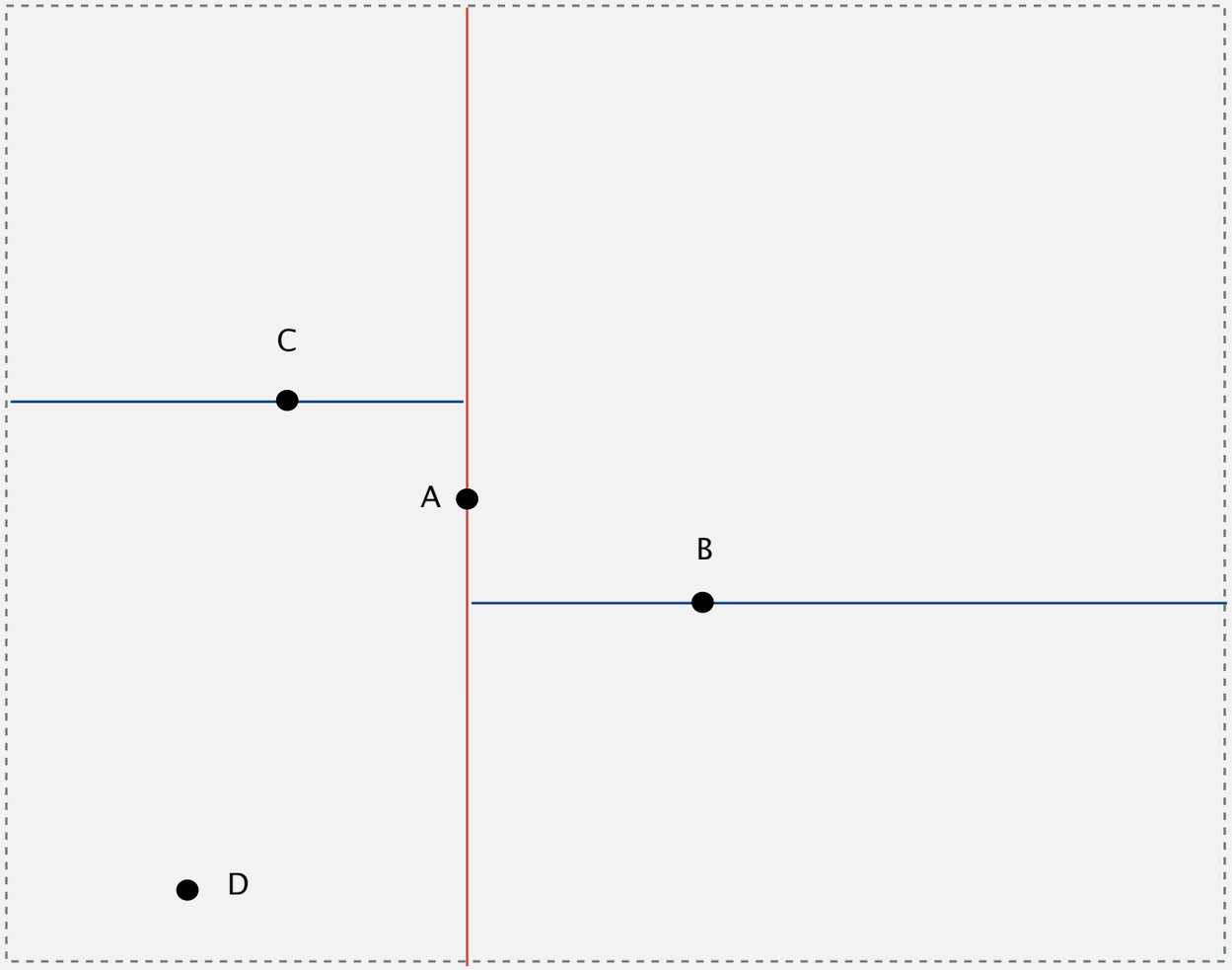
Recursively partition plane into two halfplanes.

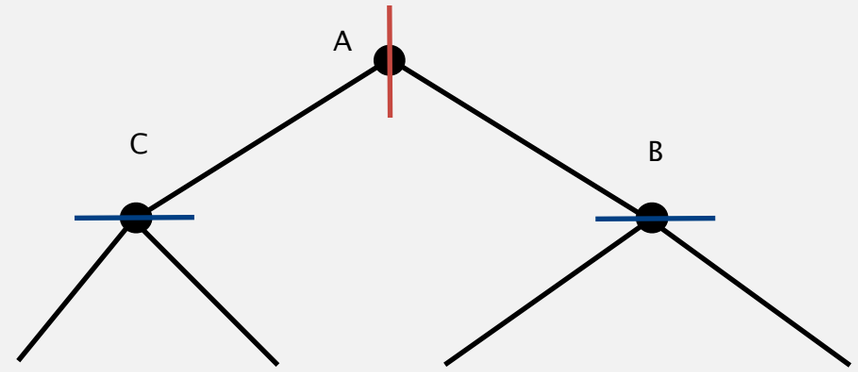




2d tree demo: insertion

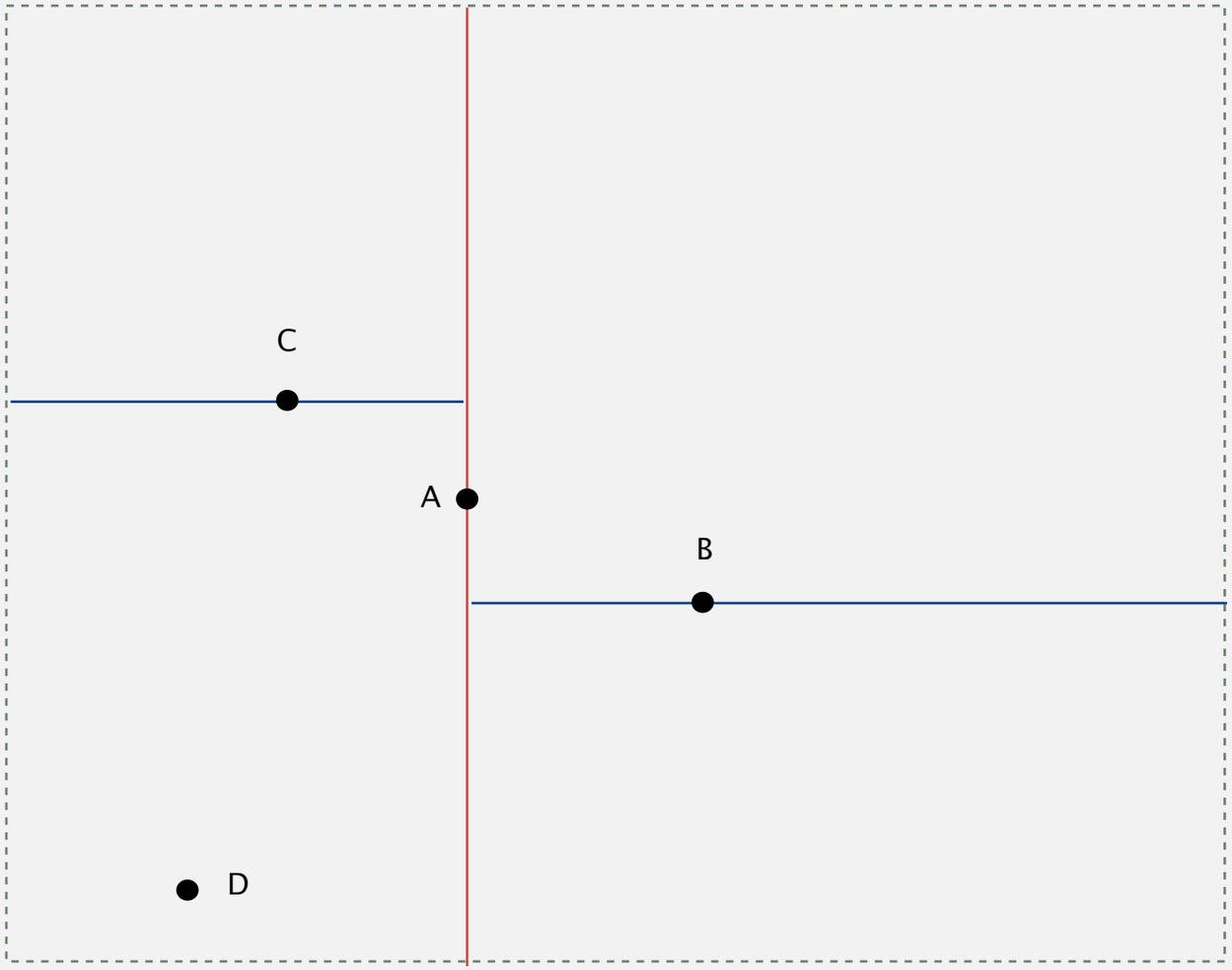
Recursively partition plane into two halfplanes.

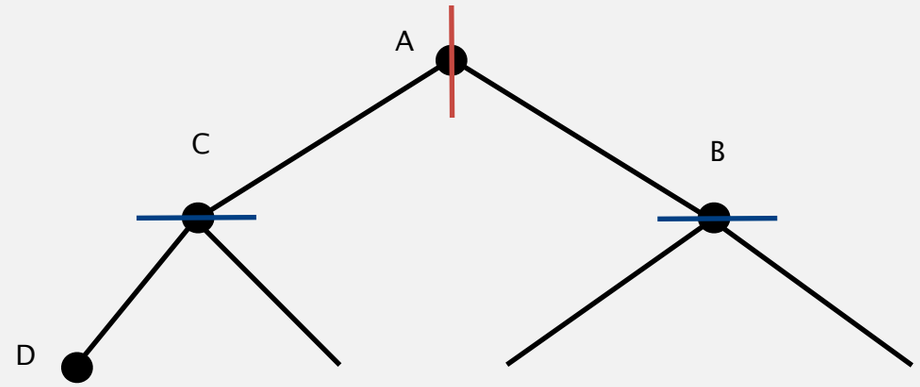




2d tree demo: insertion

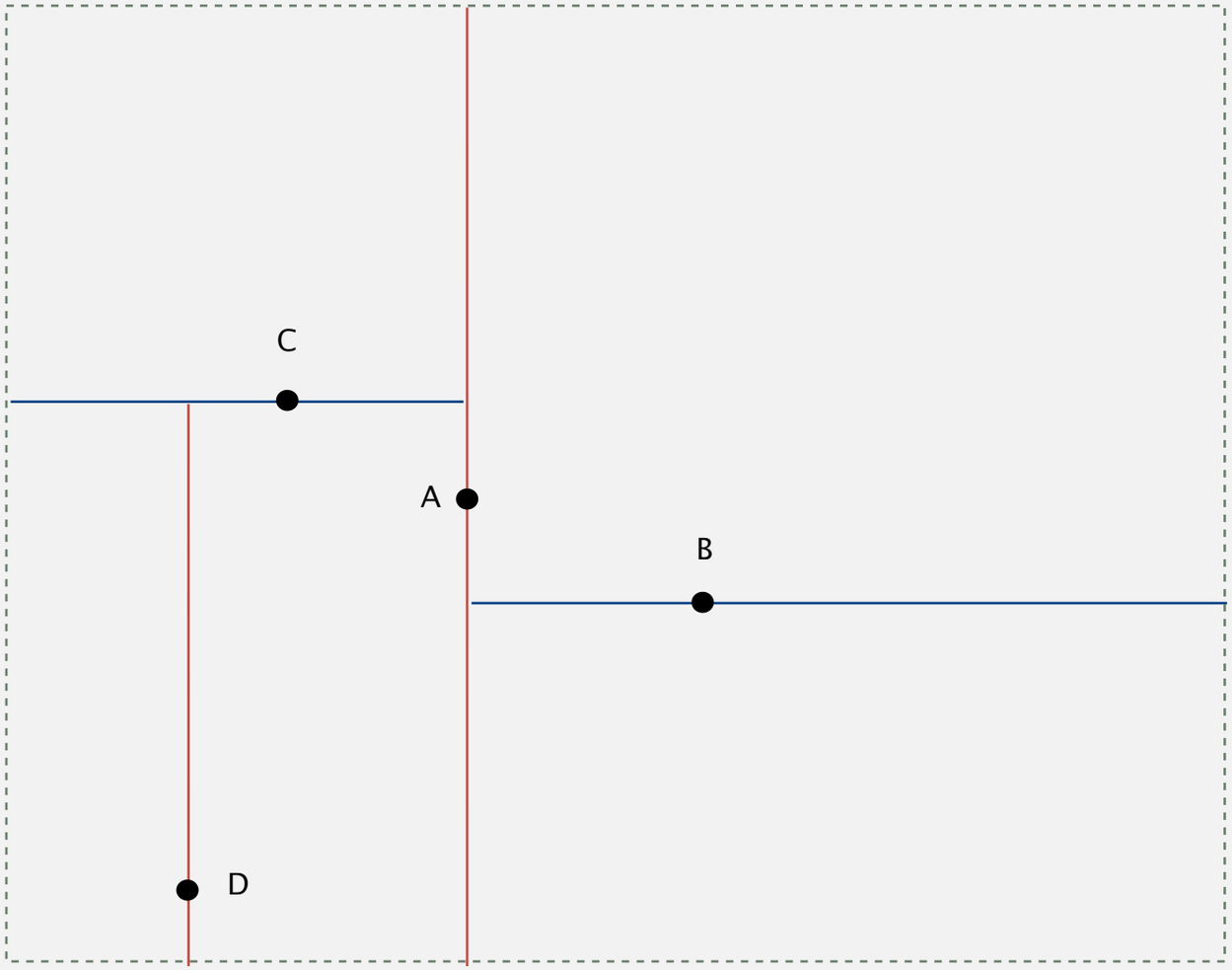
Recursively partition plane into two halfplanes.

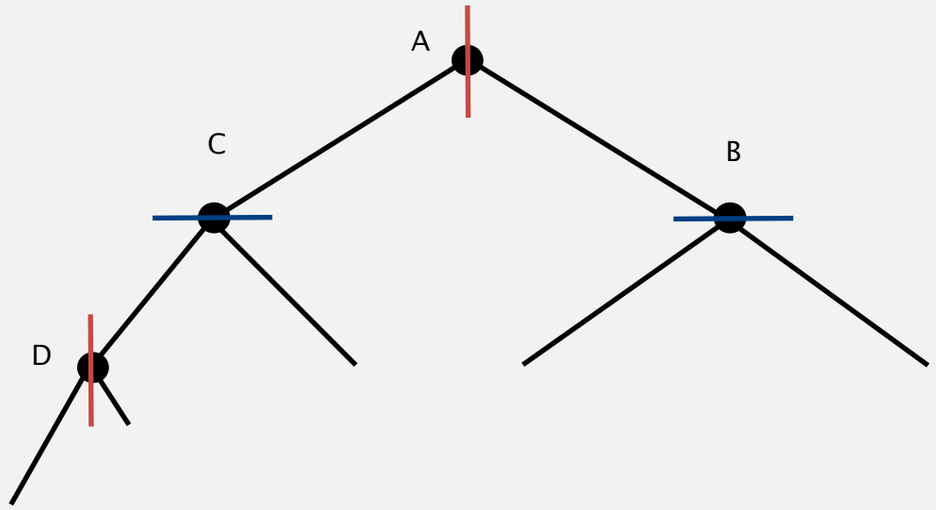




2d tree demo: insertion

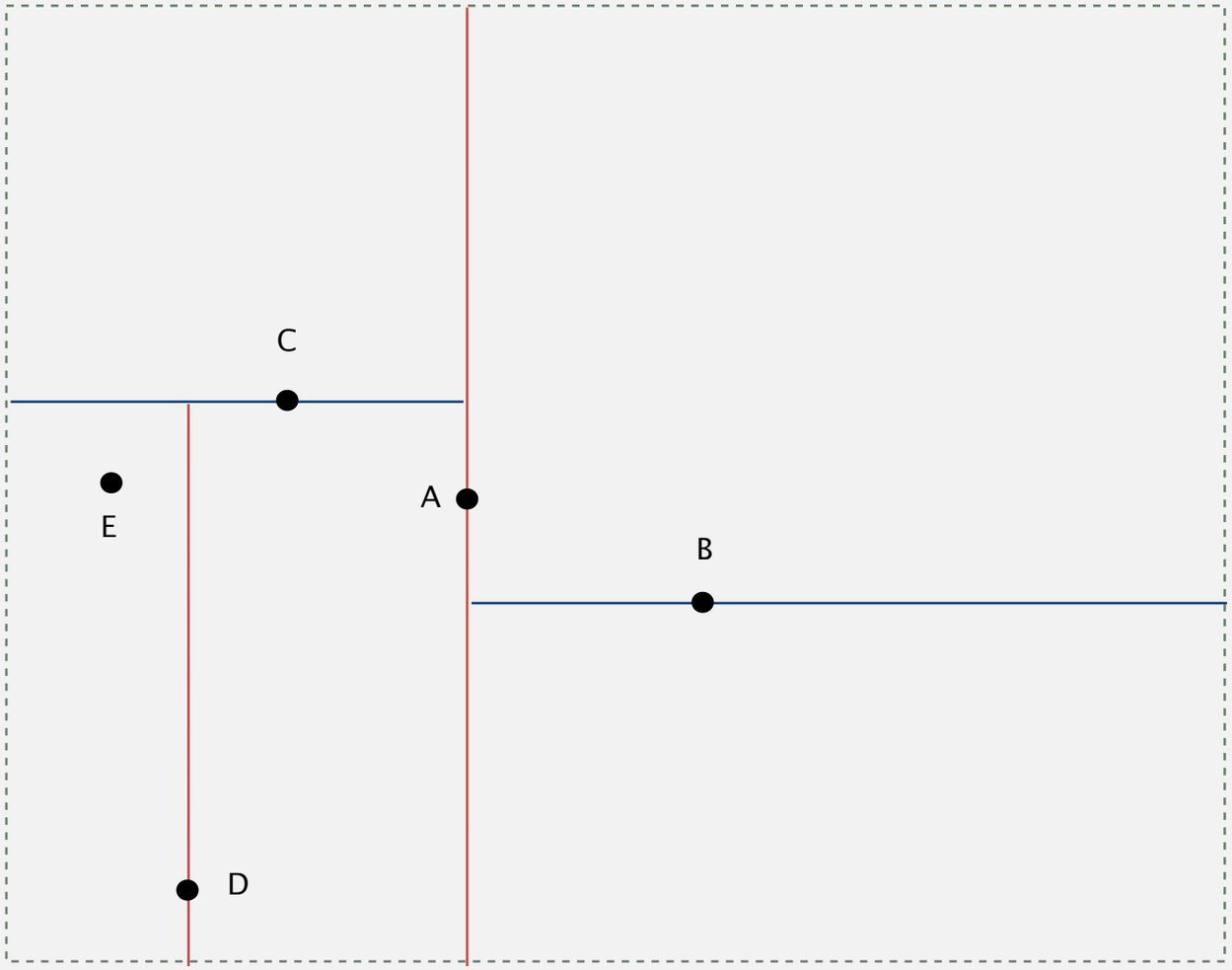
Recursively partition plane into two halfplanes.

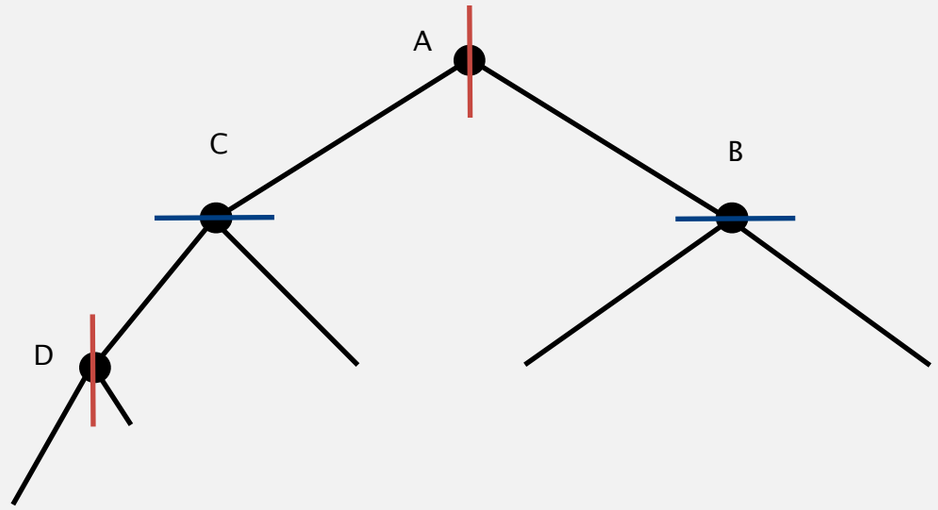




2d tree demo: insertion

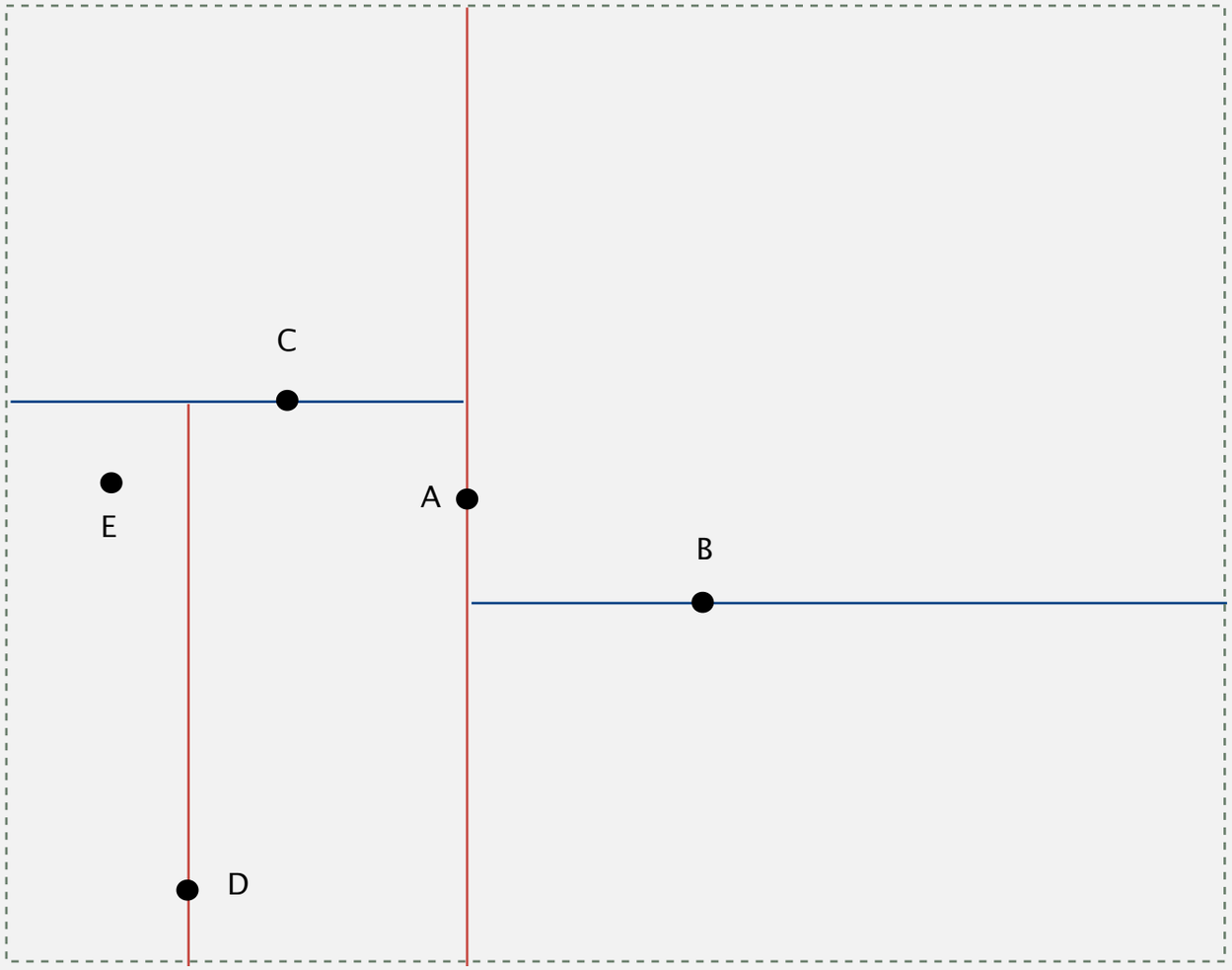
Recursively partition plane into two halfplanes.

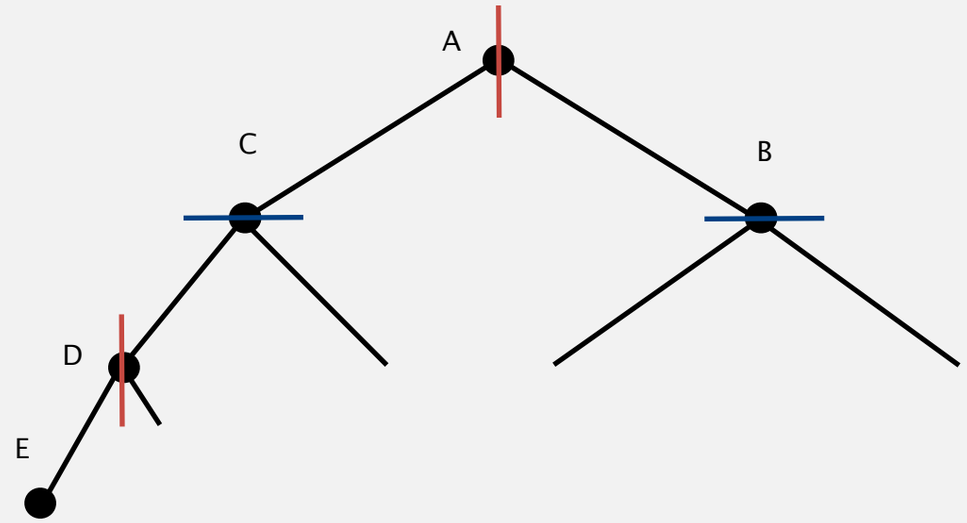




2d tree demo: insertion

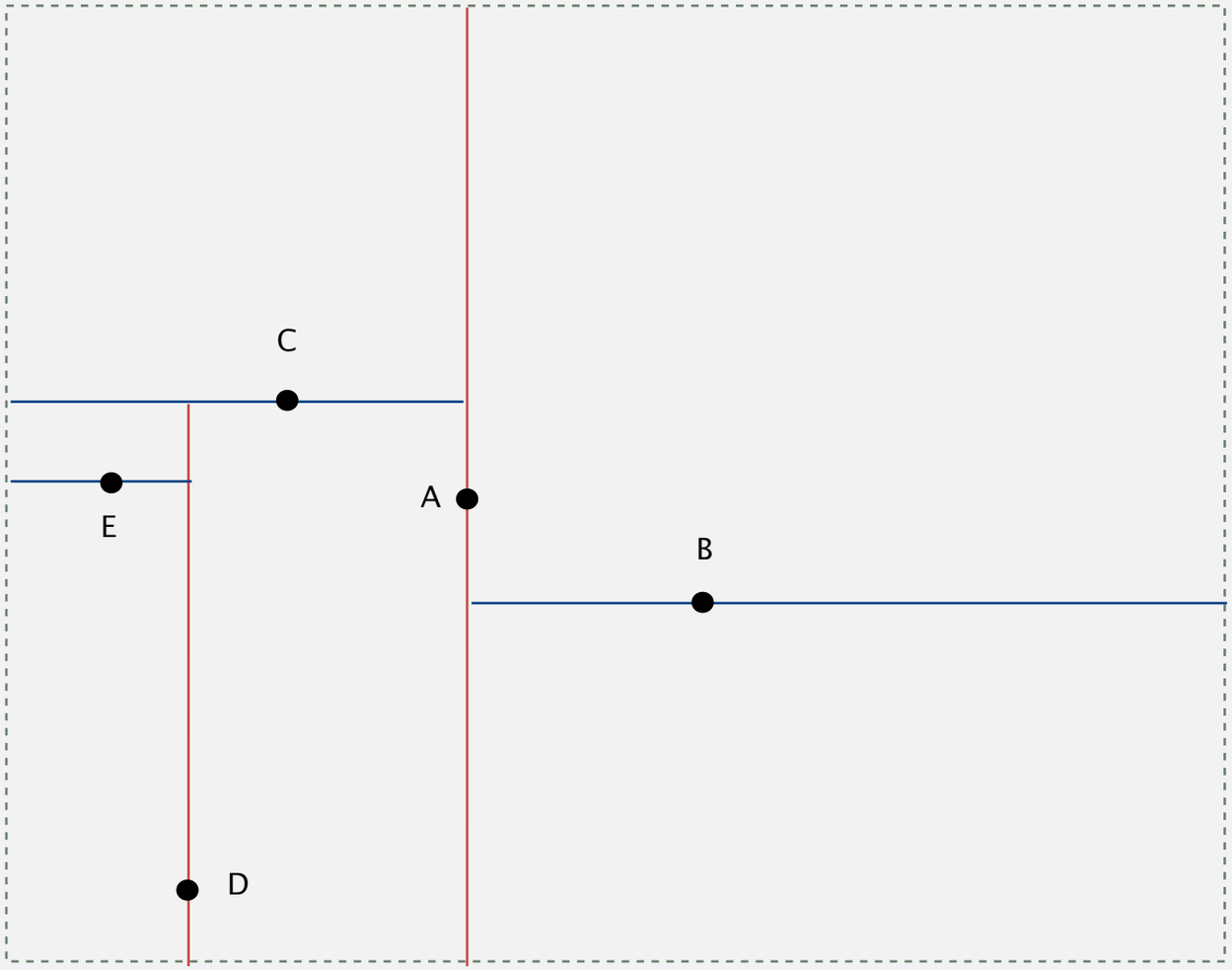
Recursively partition plane into two halfplanes.

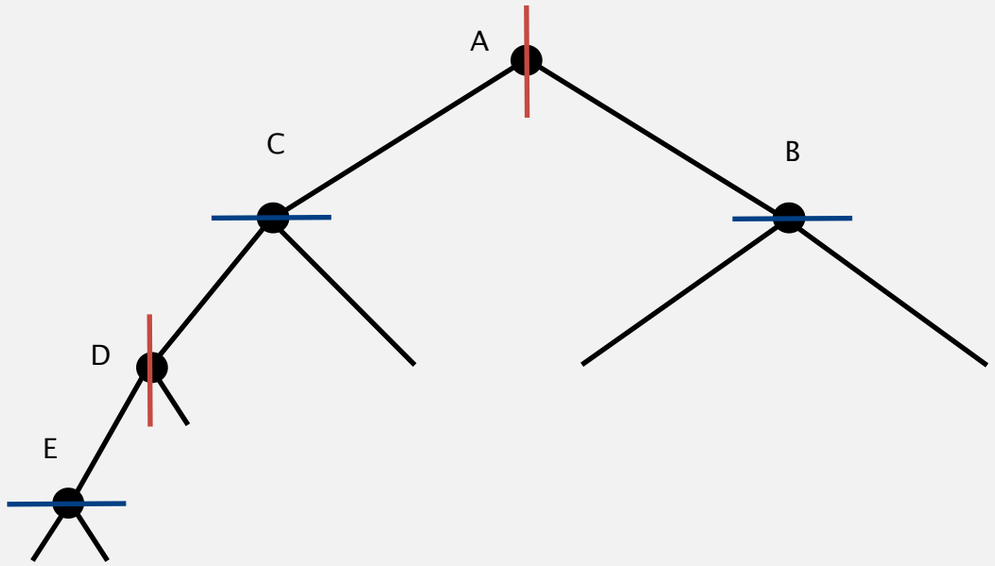




2d tree demo: insertion

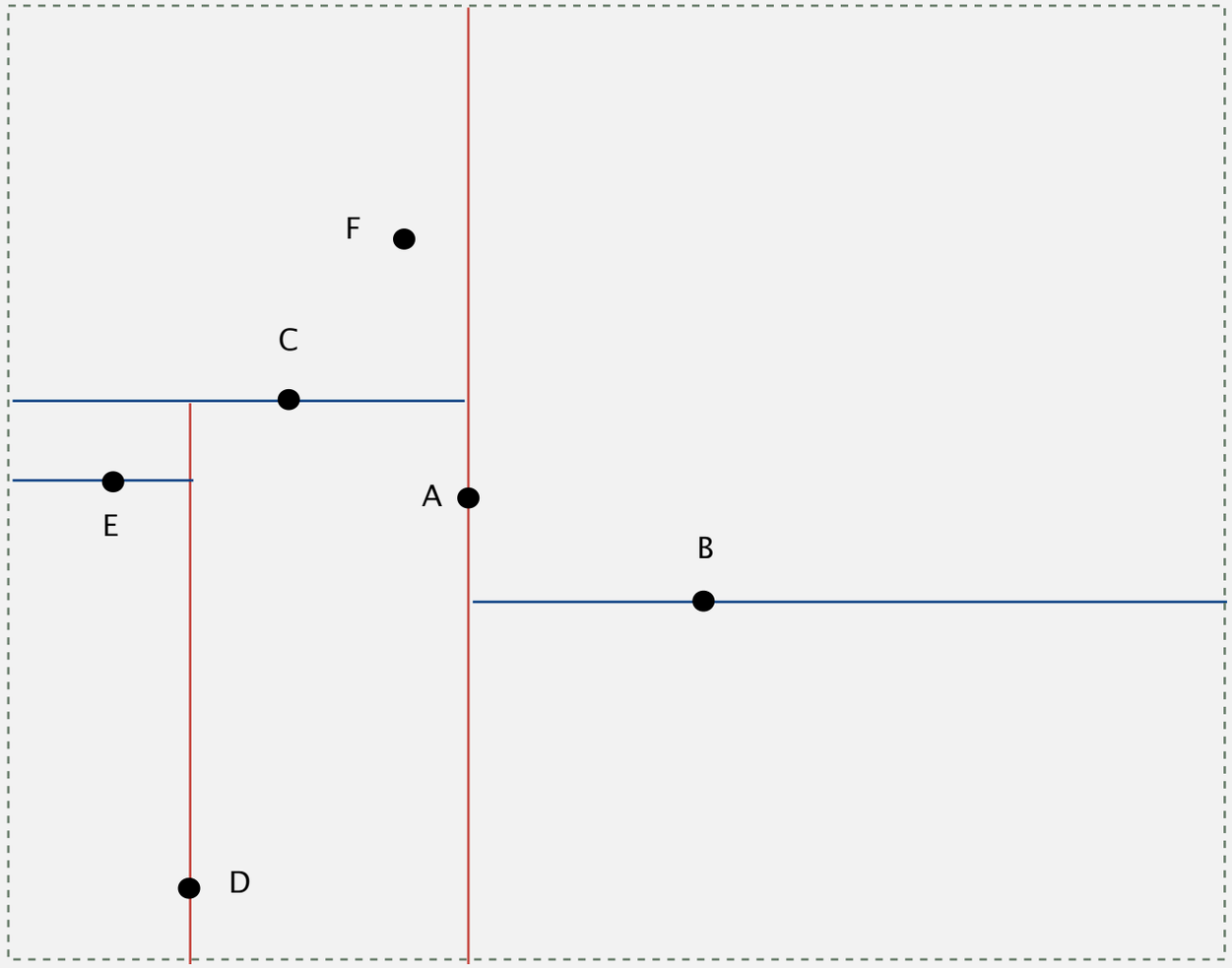
Recursively partition plane into two halfplanes.

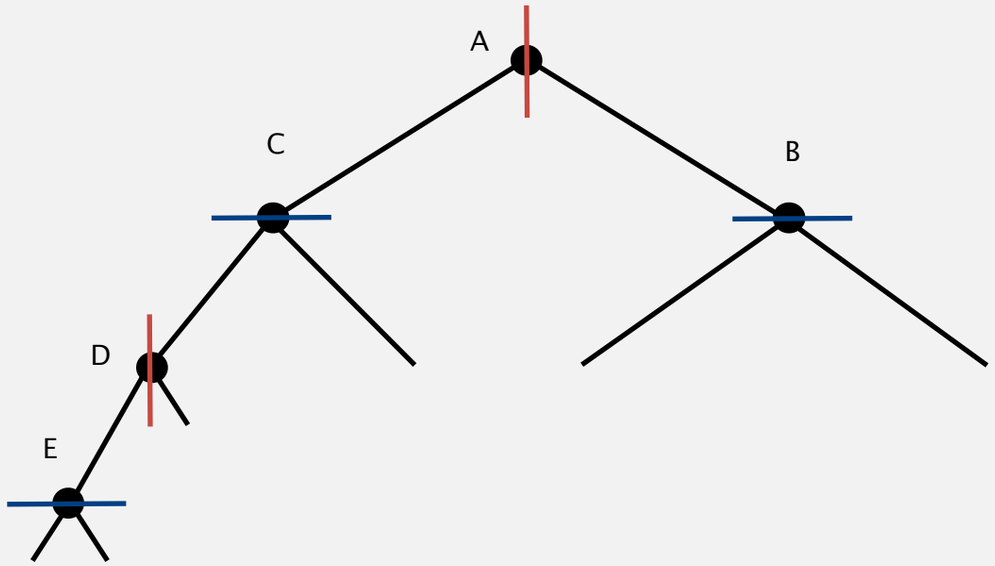




2d tree demo: insertion

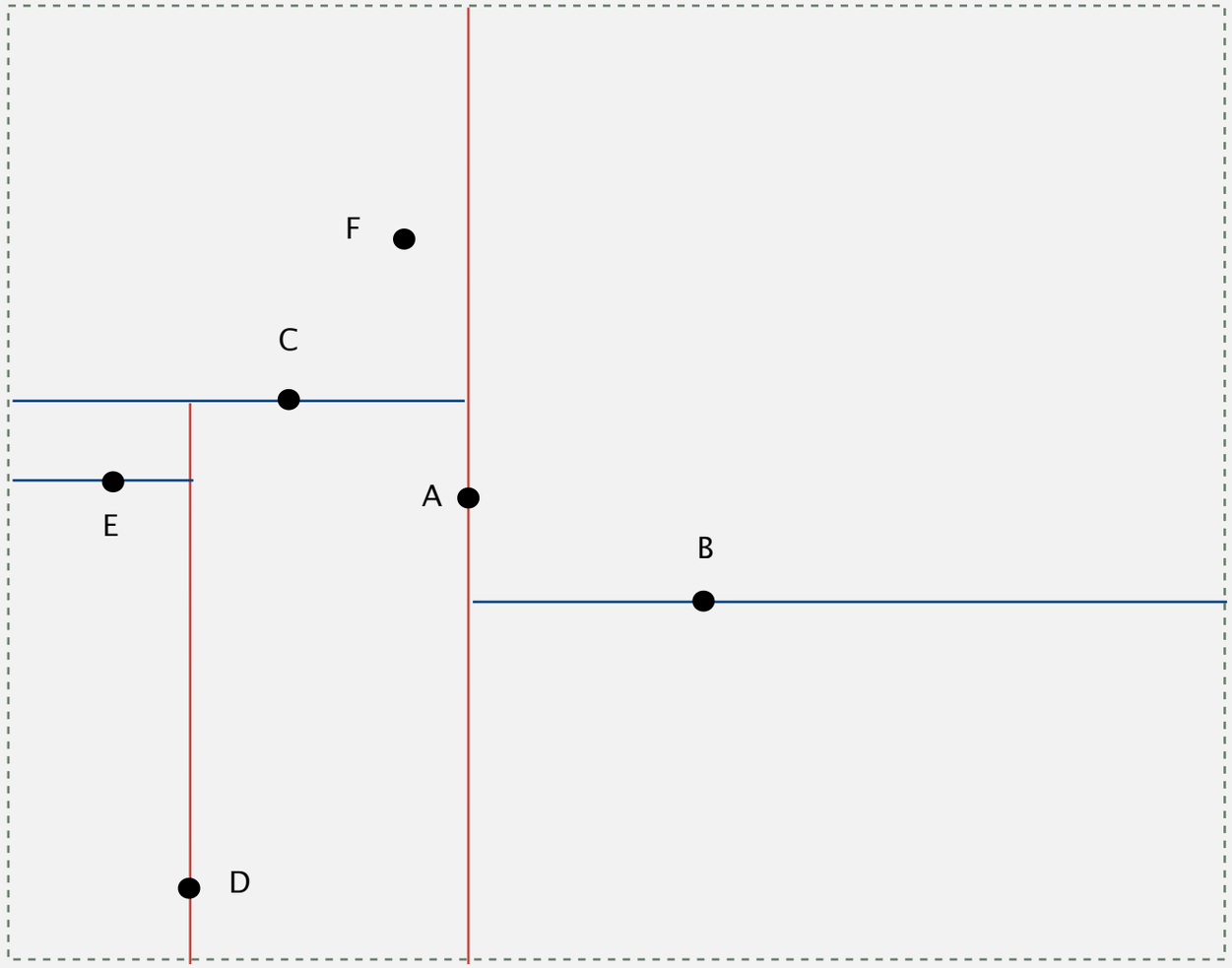
Recursively partition plane into two halfplanes.

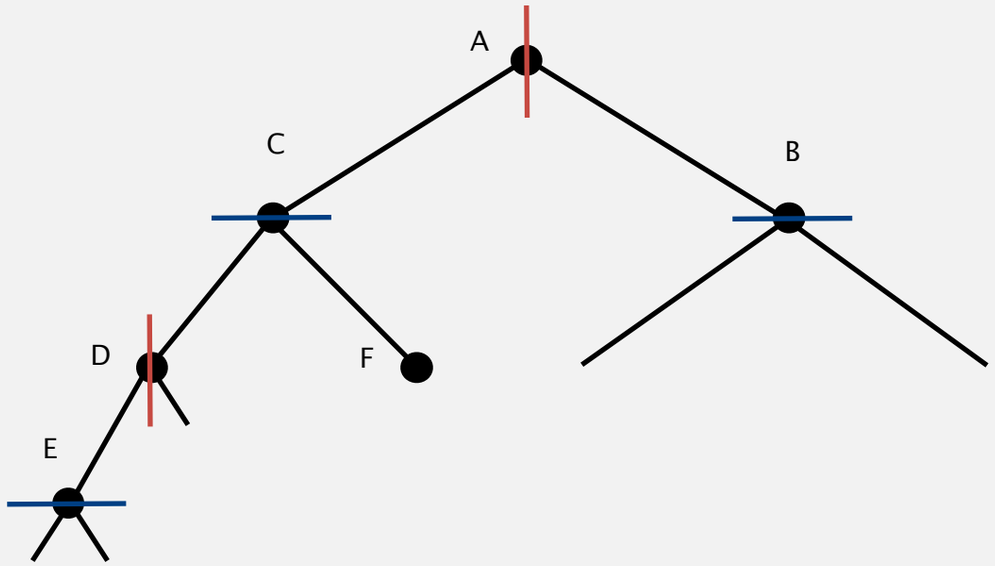




2d tree demo: insertion

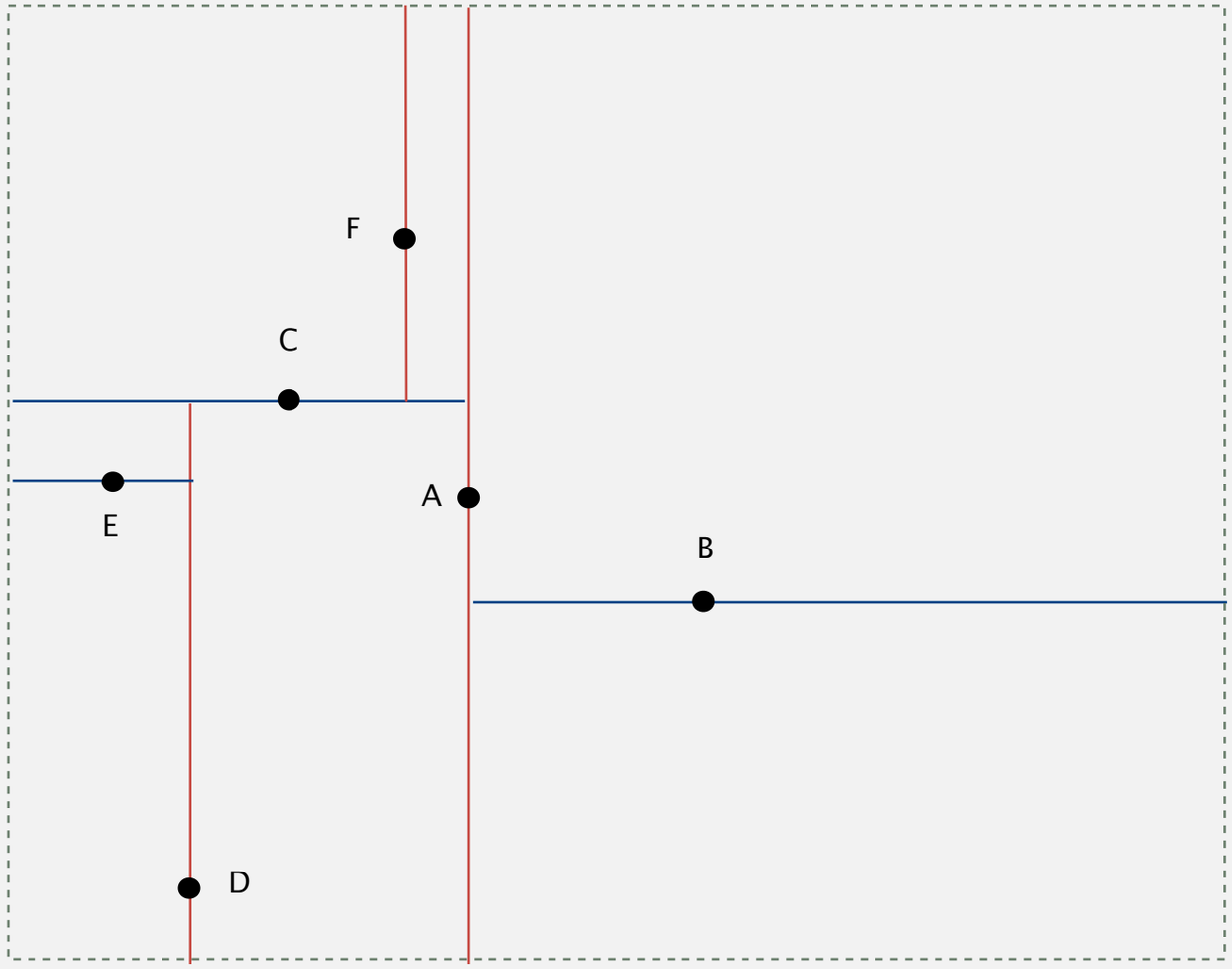
Recursively partition plane into two halfplanes.

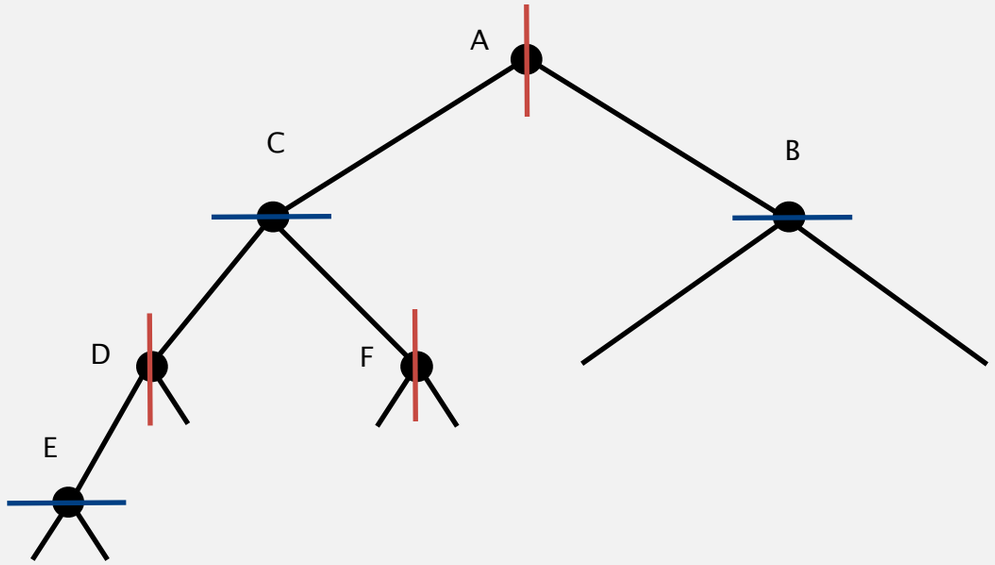




2d tree demo: insertion

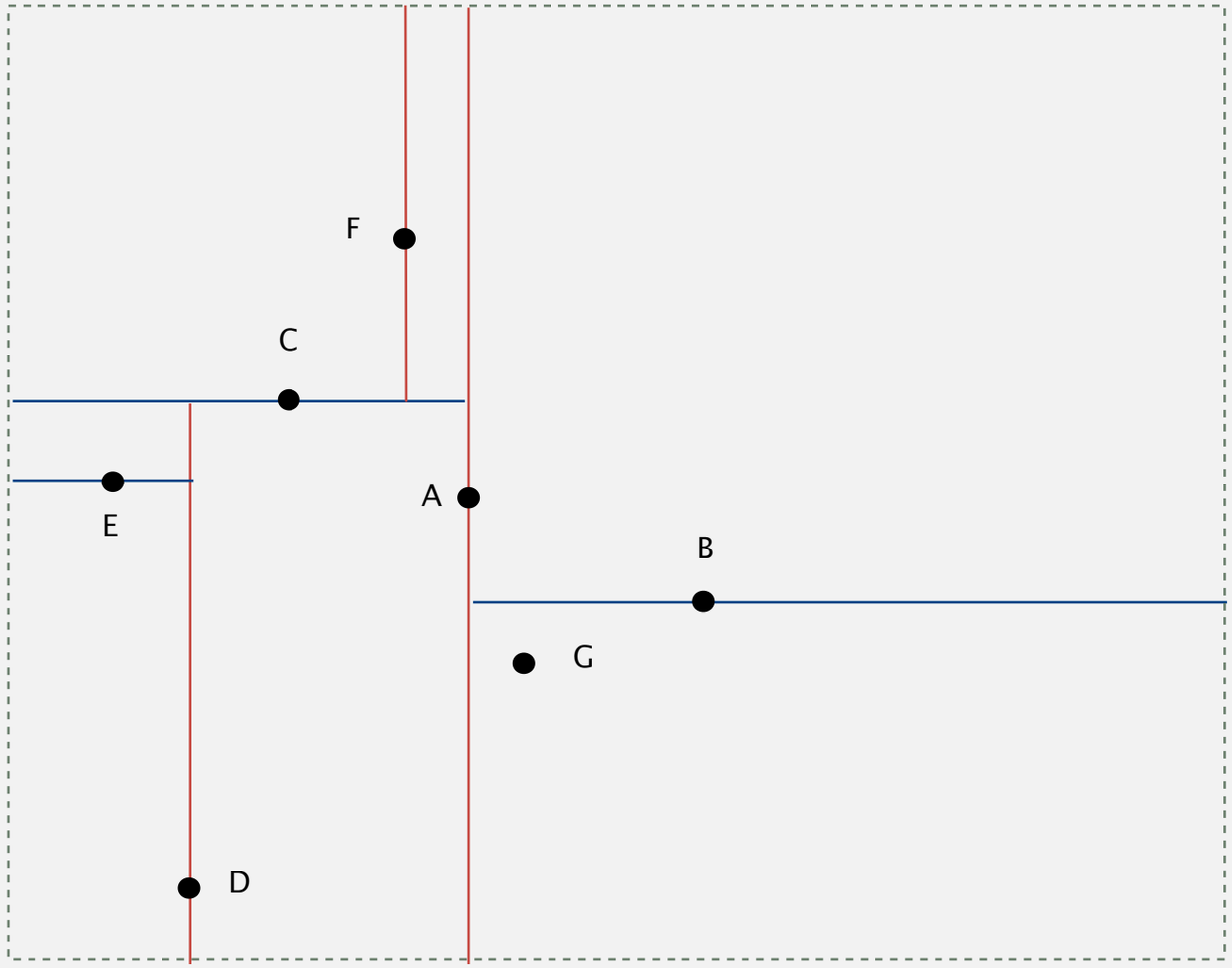
Recursively partition plane into two halfplanes.

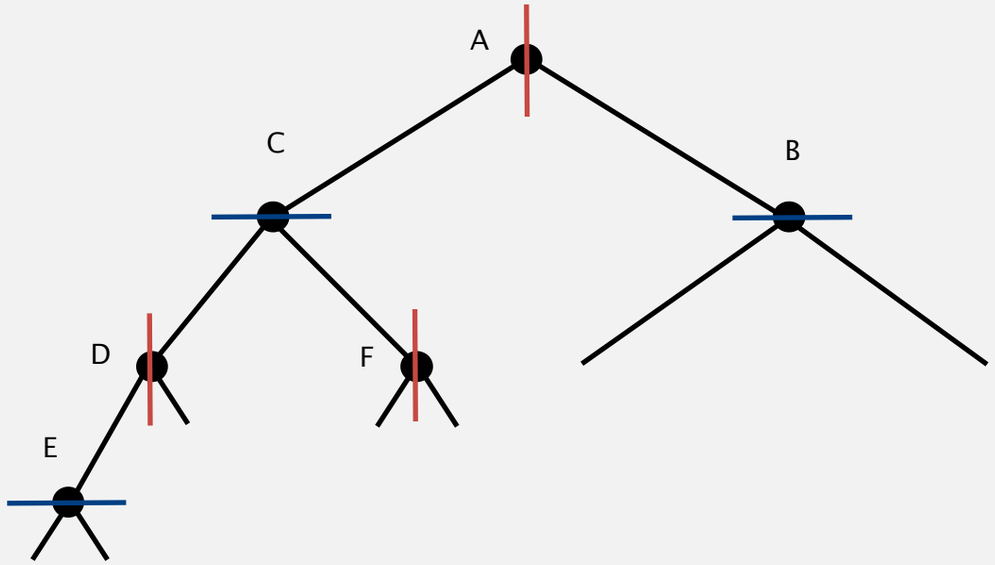




2d tree demo: insertion

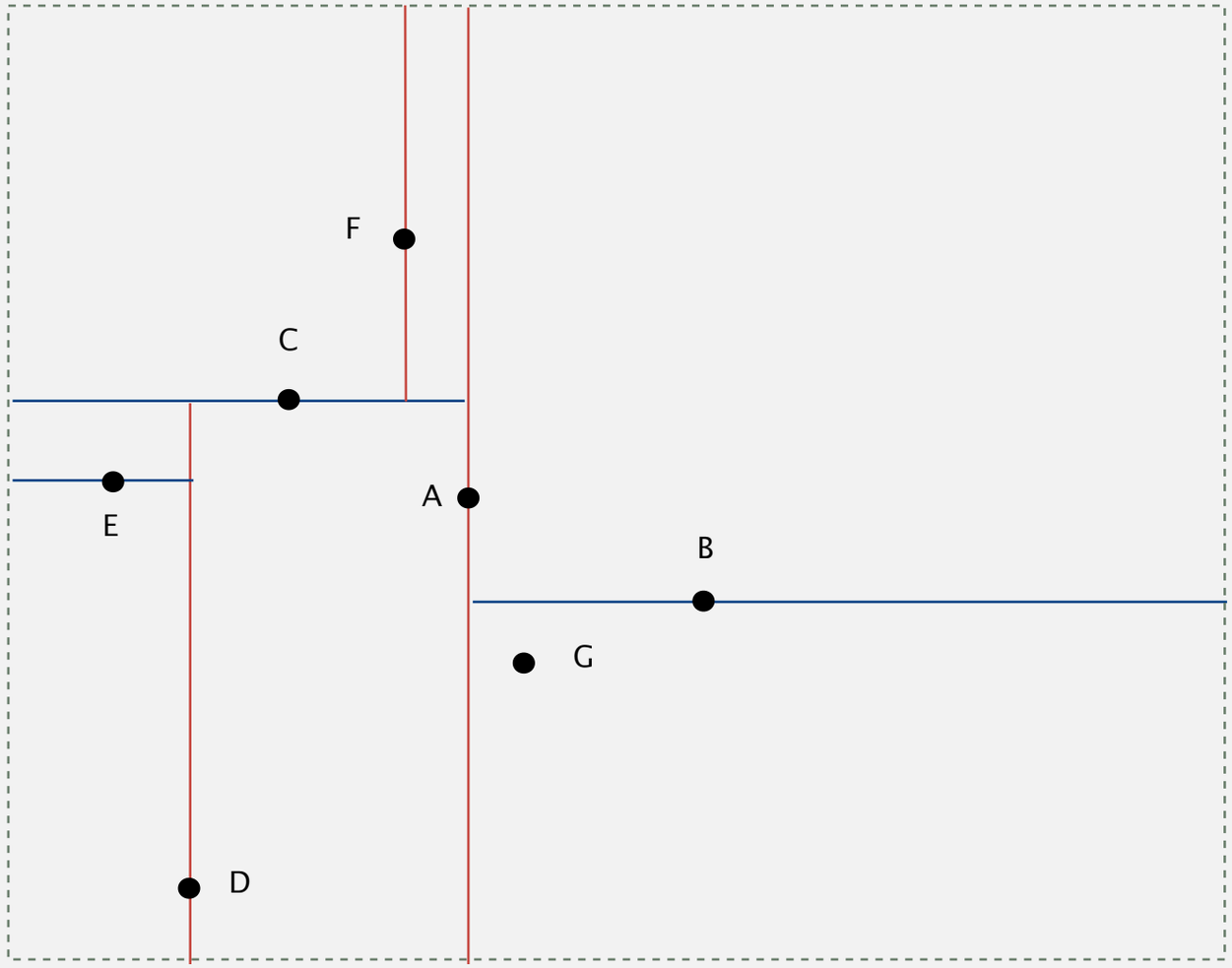
Recursively partition plane into two halfplanes.

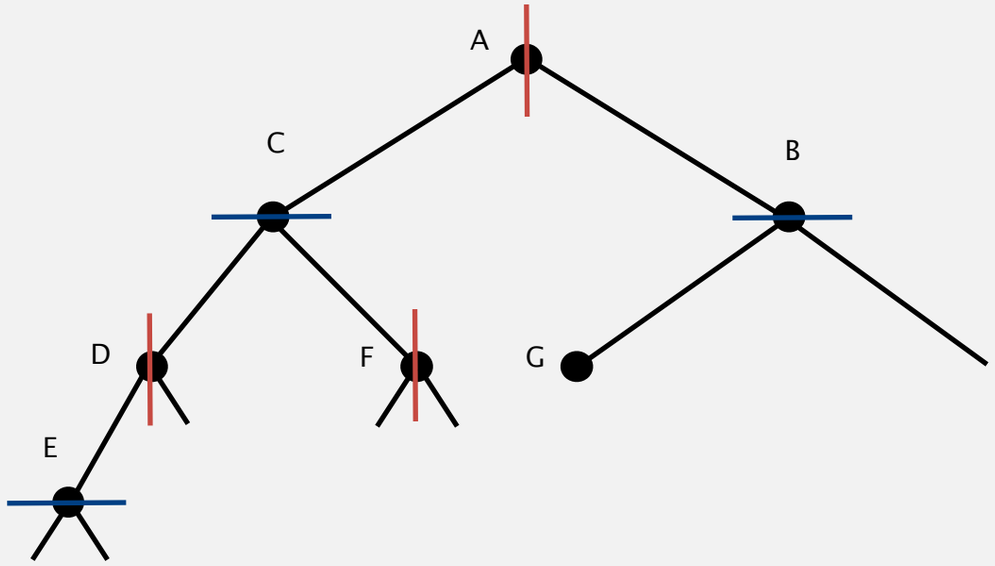




2d tree demo: insertion

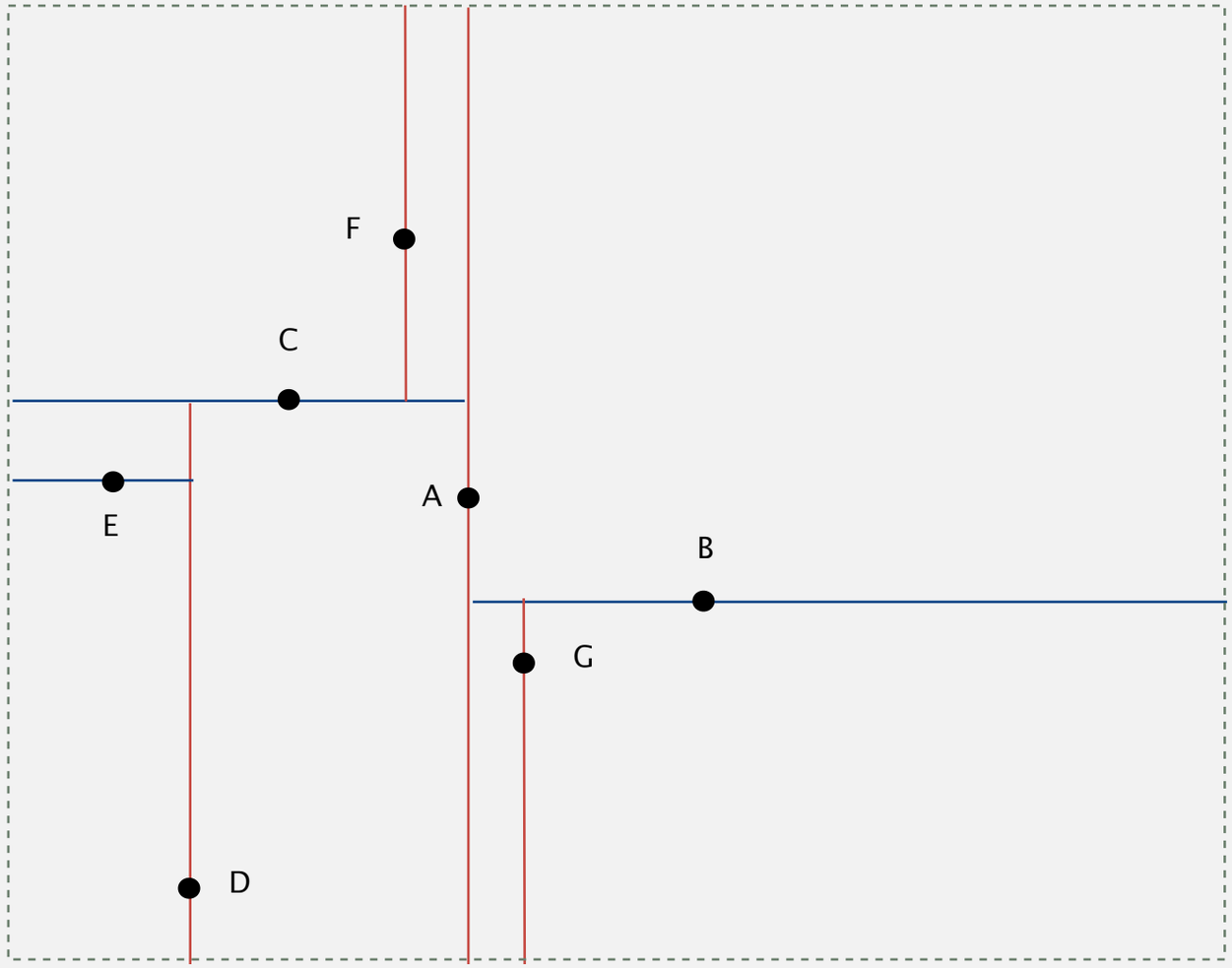
Recursively partition plane into two halfplanes.

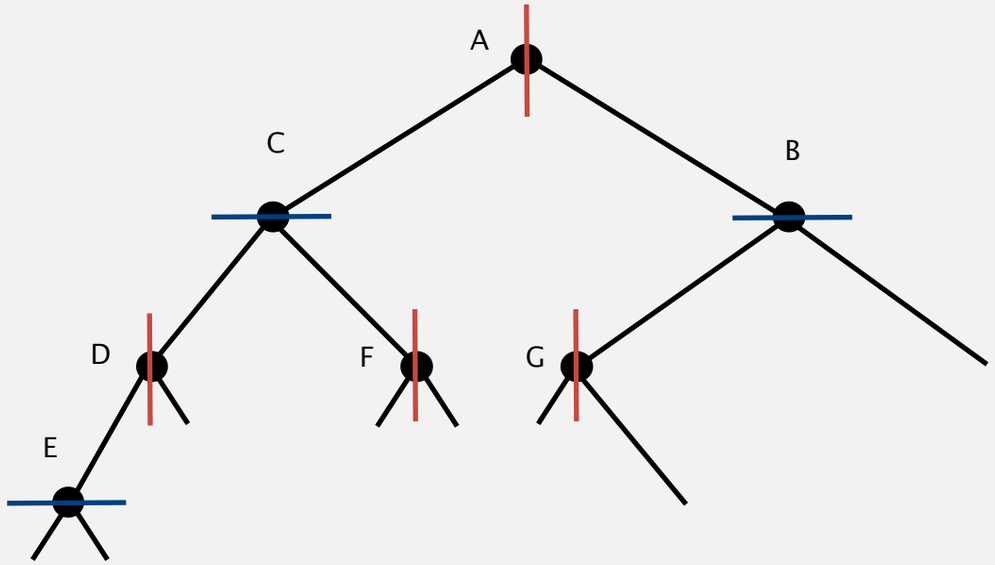




2d tree demo: insertion

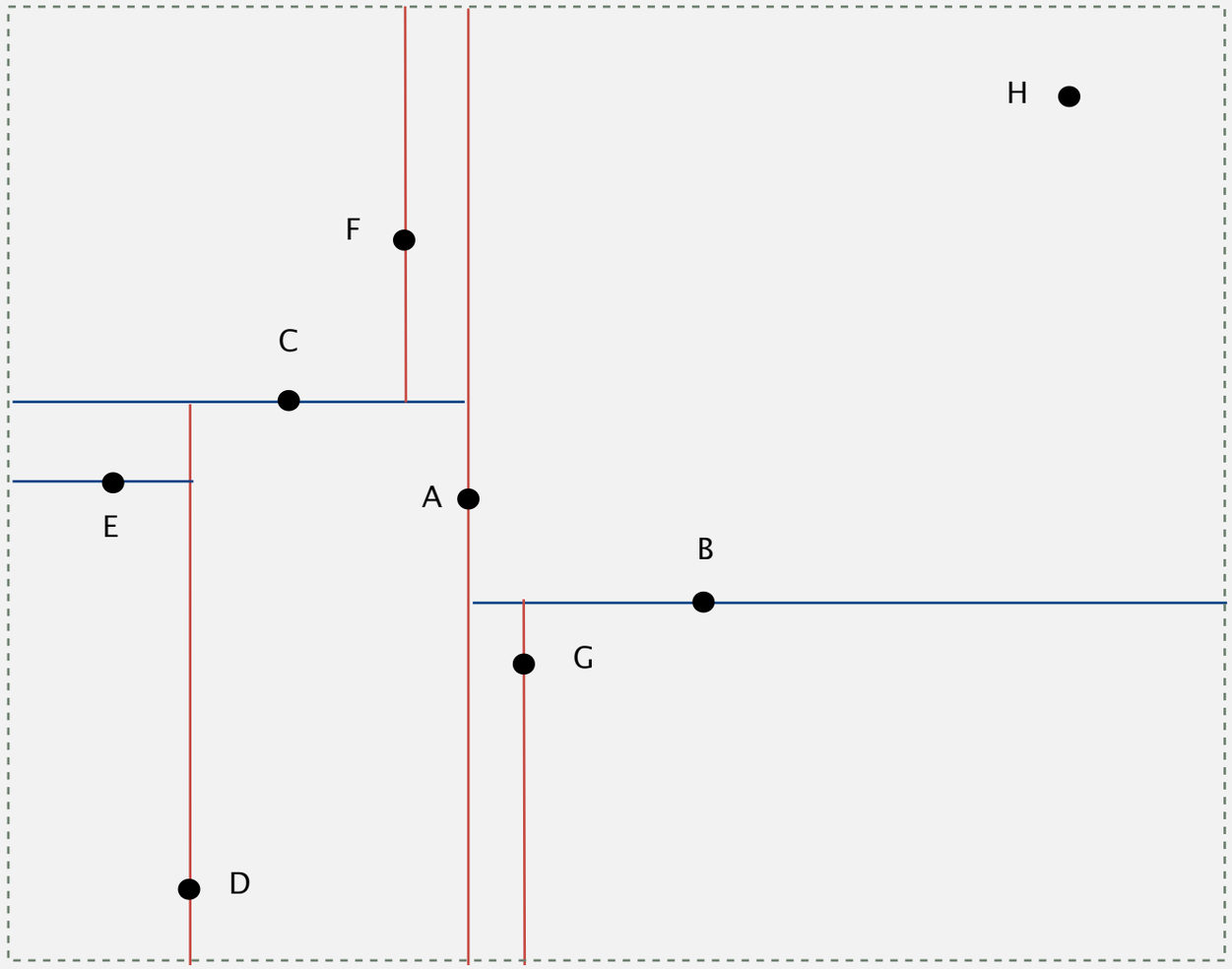
Recursively partition plane into two halfplanes.

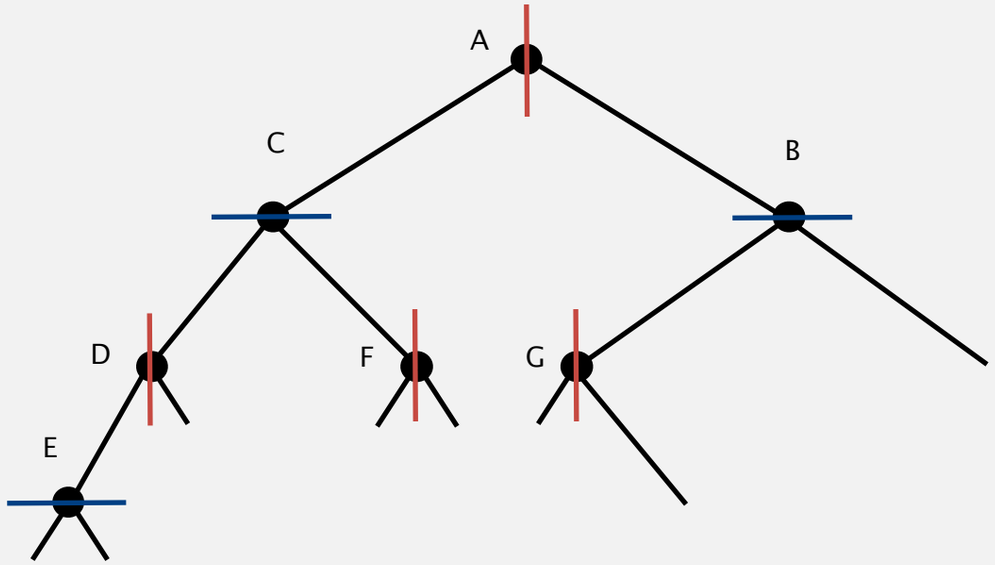




2d tree demo: insertion

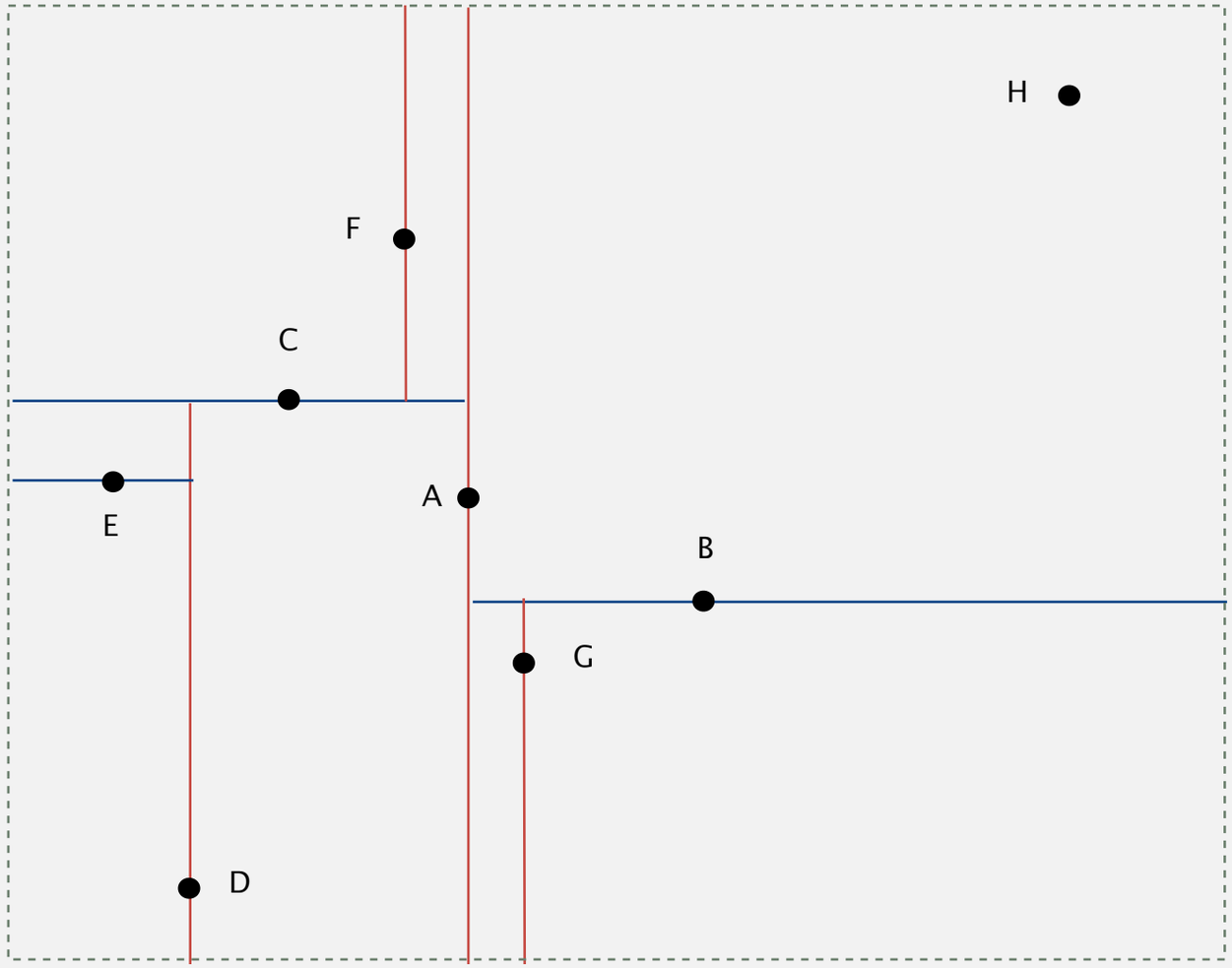
Recursively partition plane into two halfplanes.

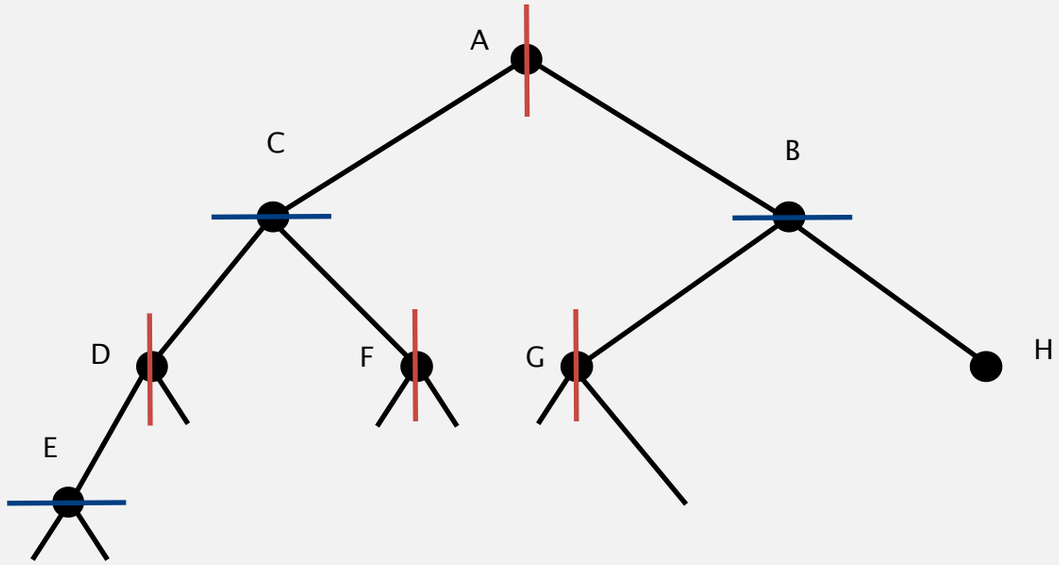




2d tree demo: insertion

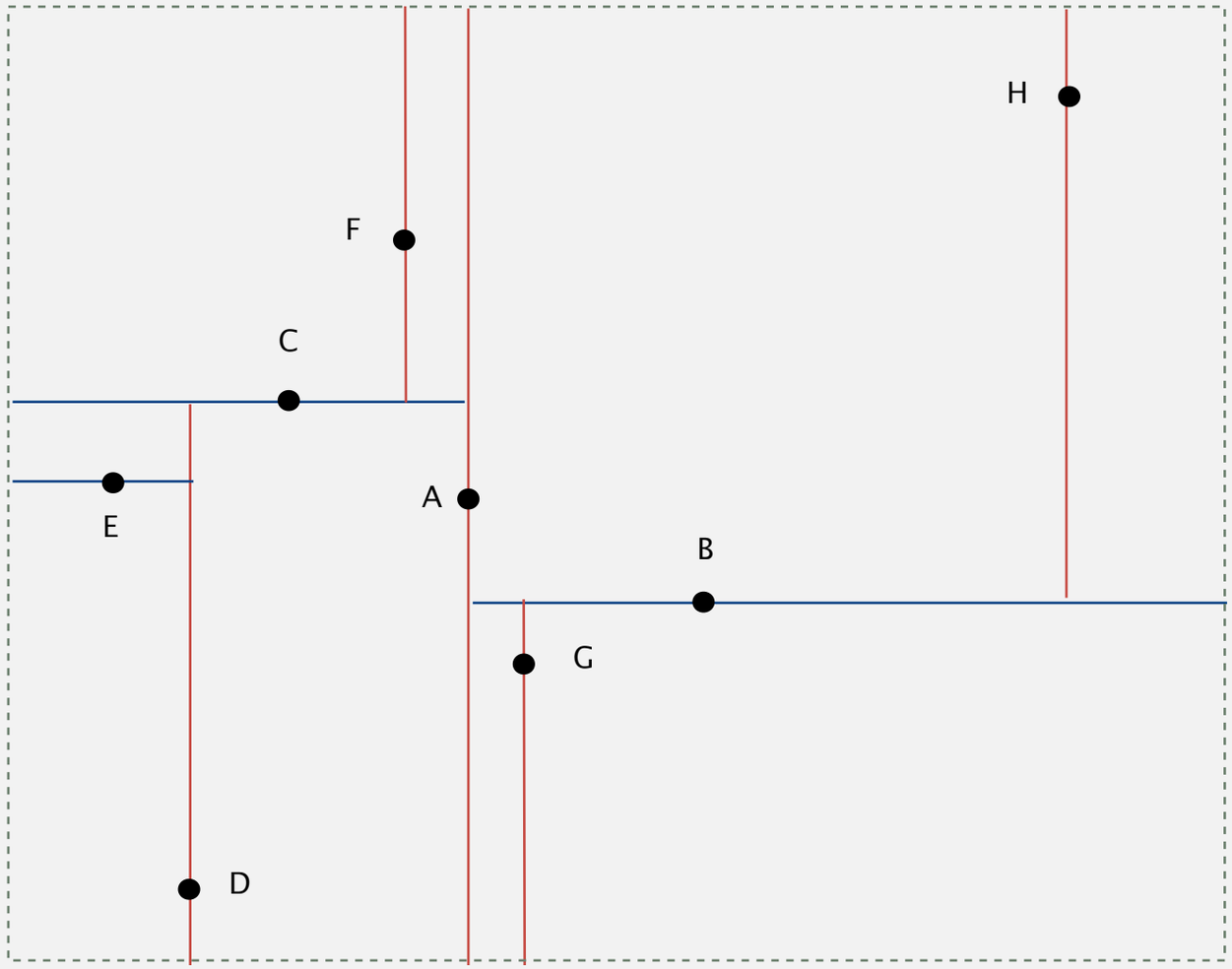
Recursively partition plane into two halfplanes.

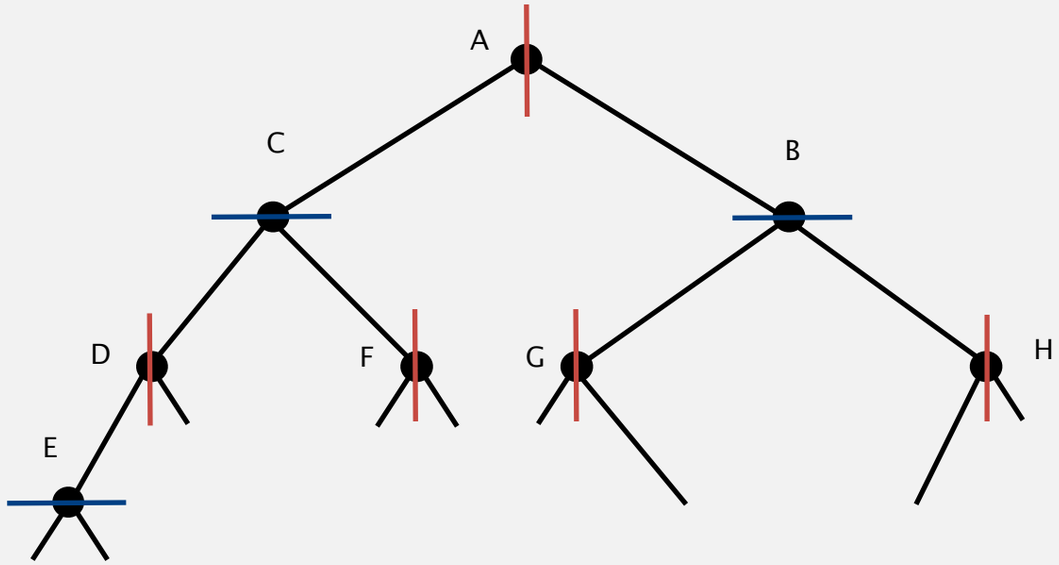




2d tree demo: insertion

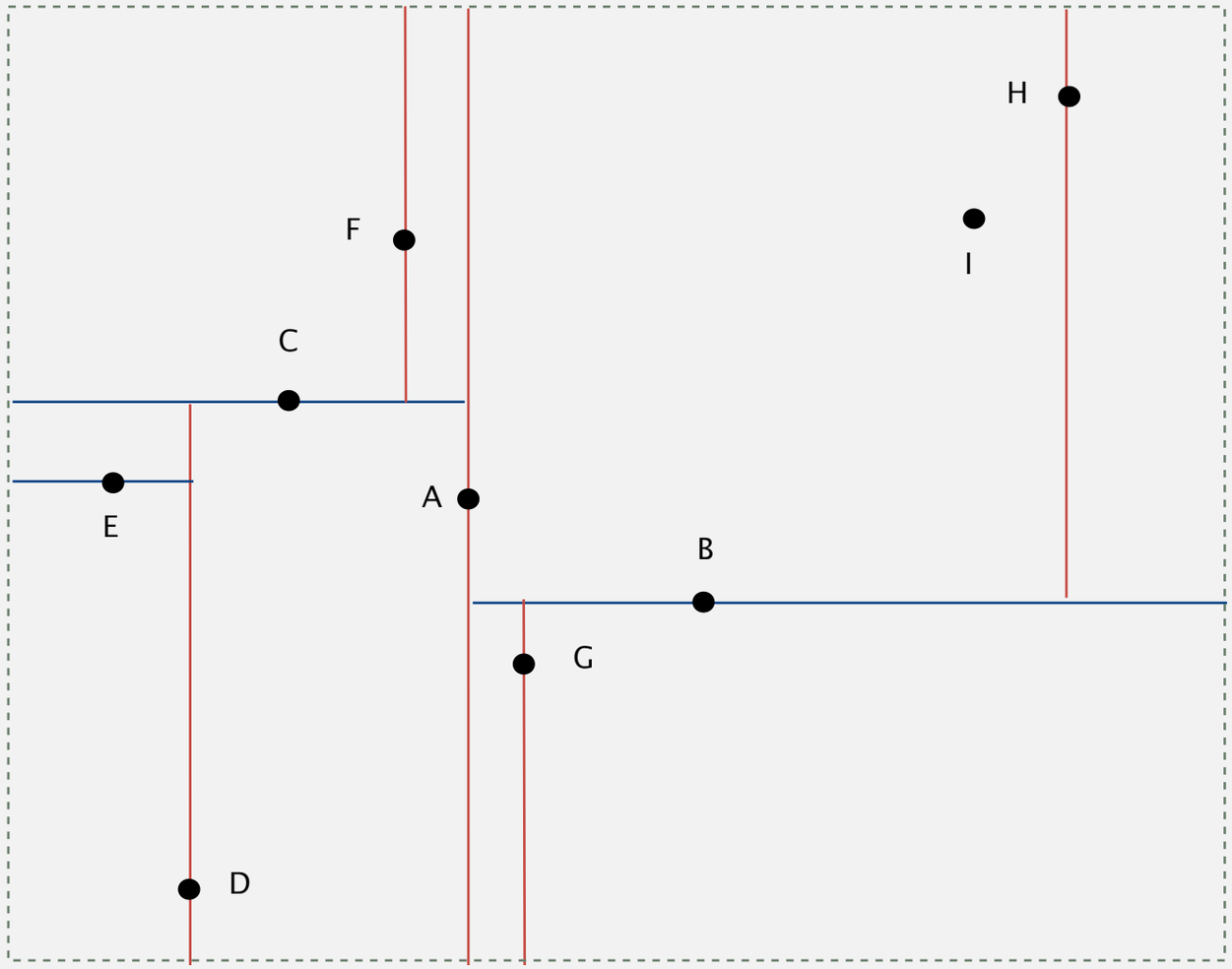
Recursively partition plane into two halfplanes.

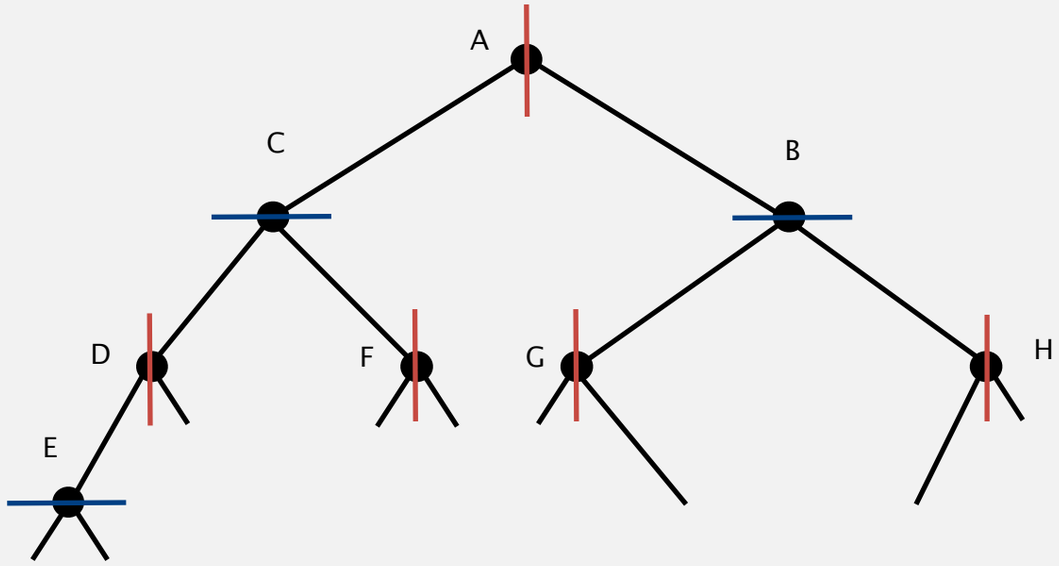




2d tree demo: insertion

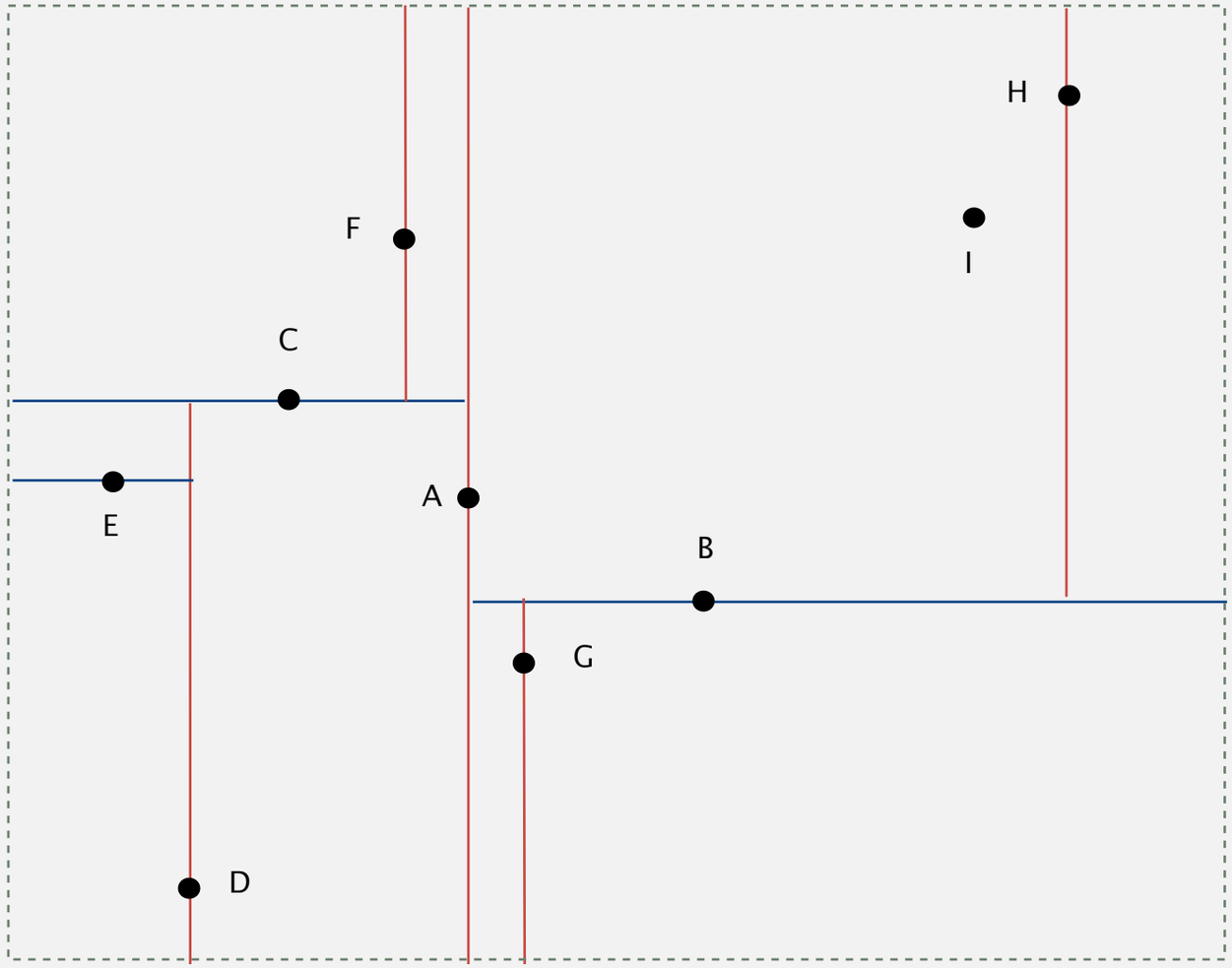
Recursively partition plane into two halfplanes.

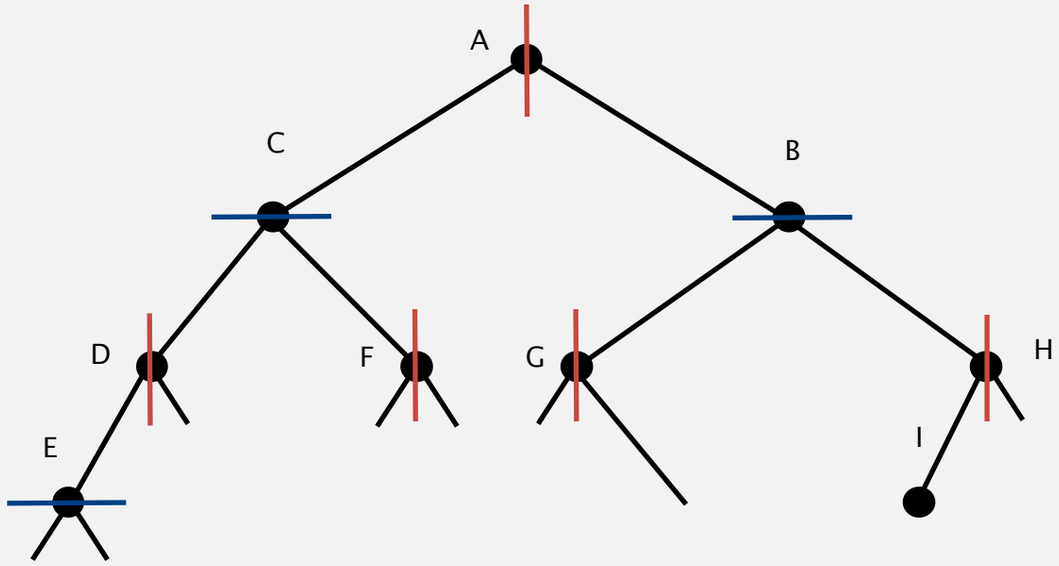




2d tree demo: insertion

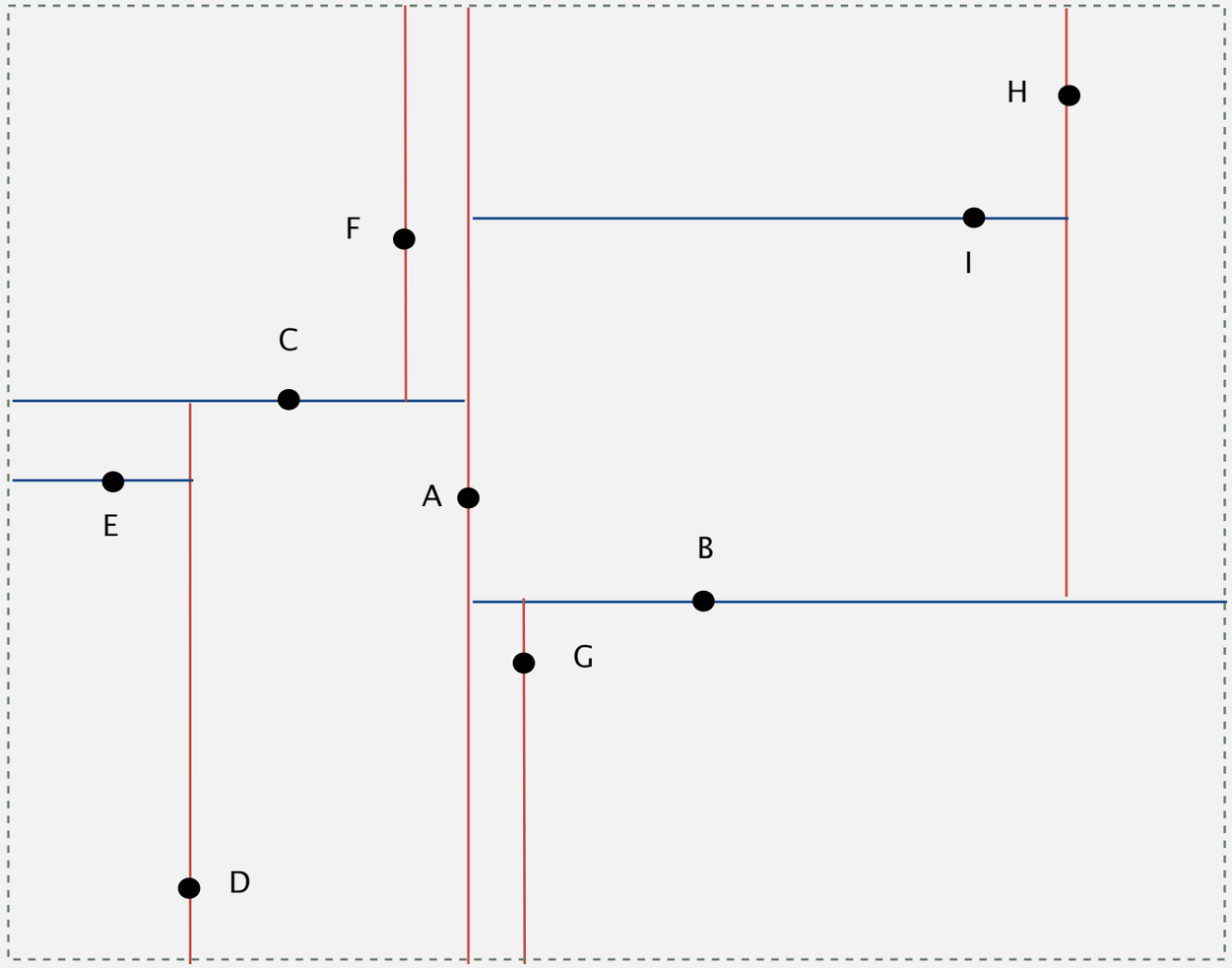
Recursively partition plane into two halfplanes.

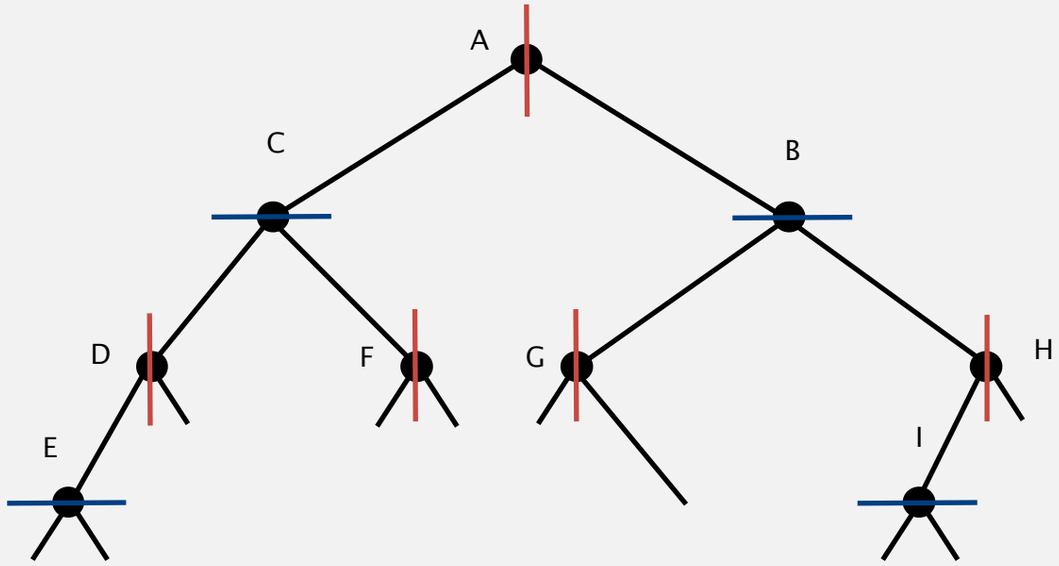




2d tree demo: insertion

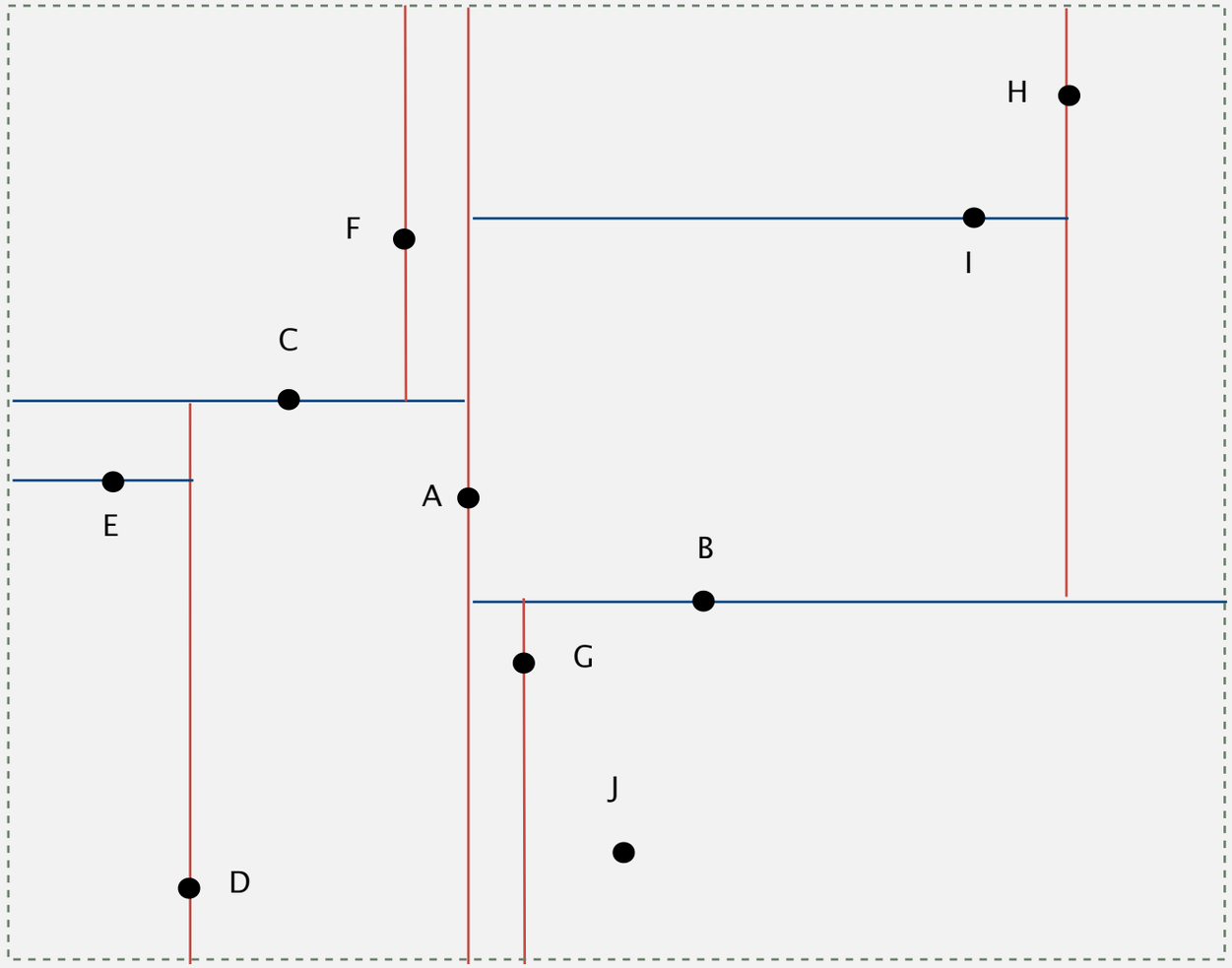
Recursively partition plane into two halfplanes.

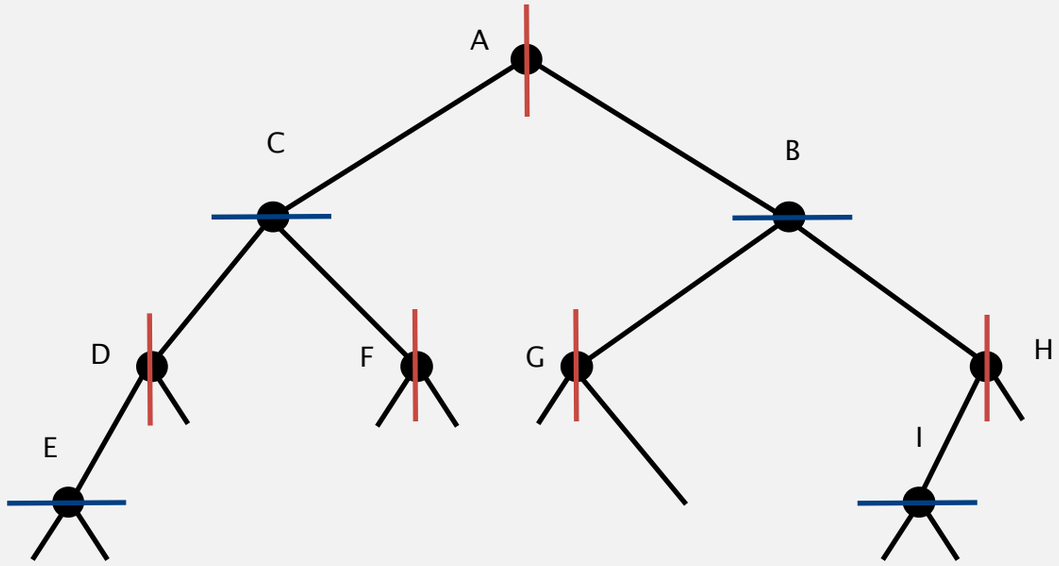




2d tree demo: insertion

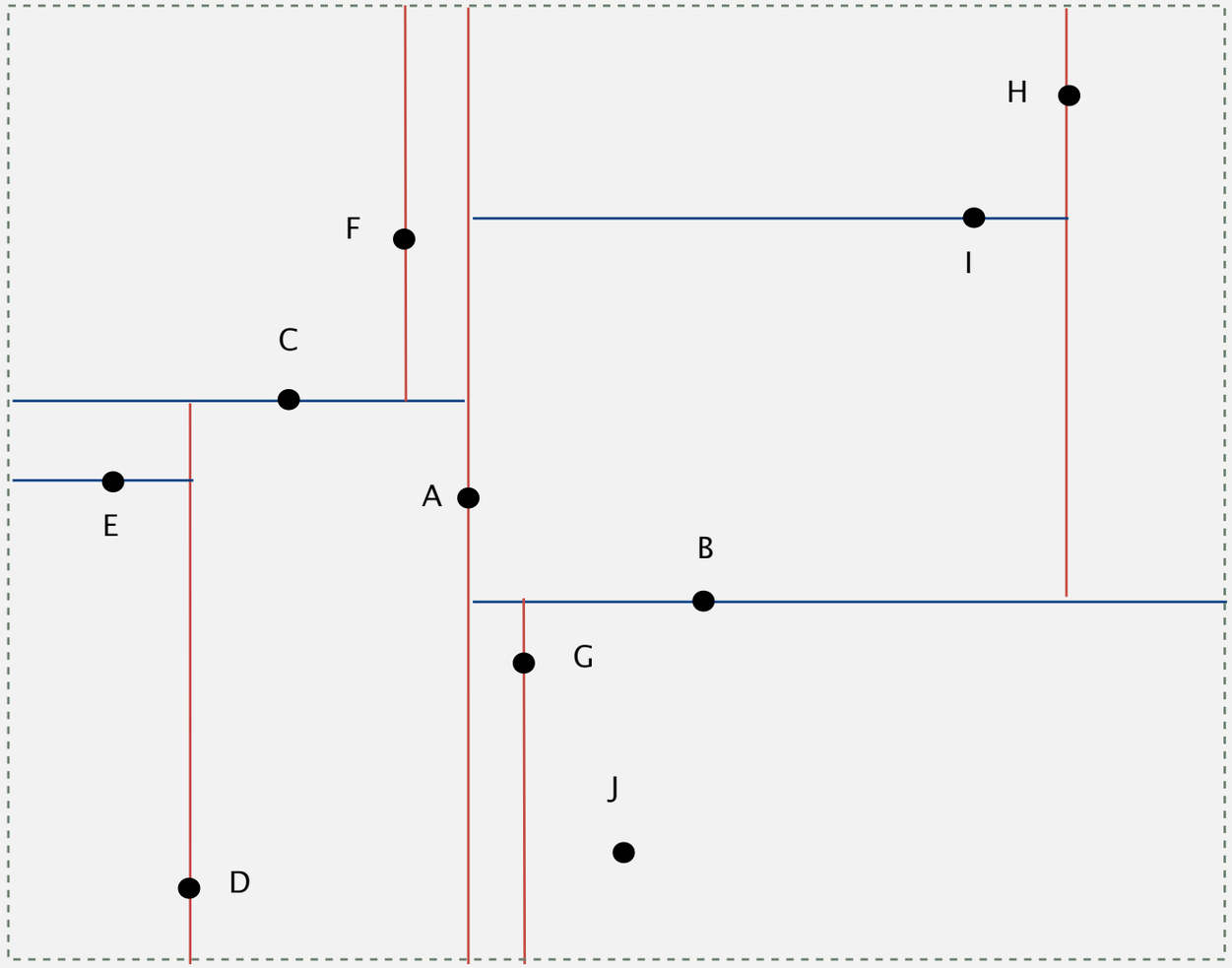
Recursively partition plane into two halfplanes.

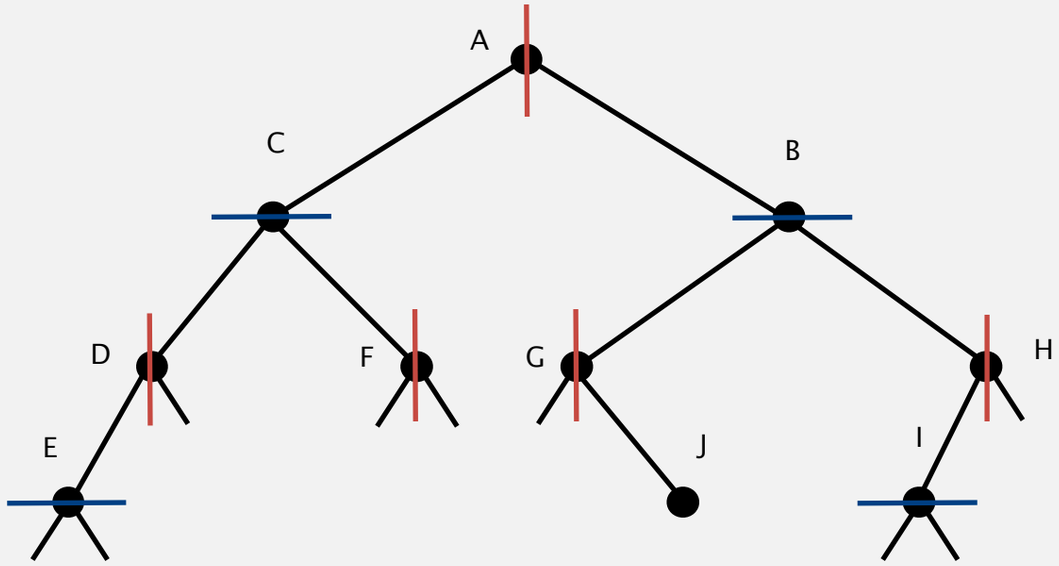




2d tree demo: insertion

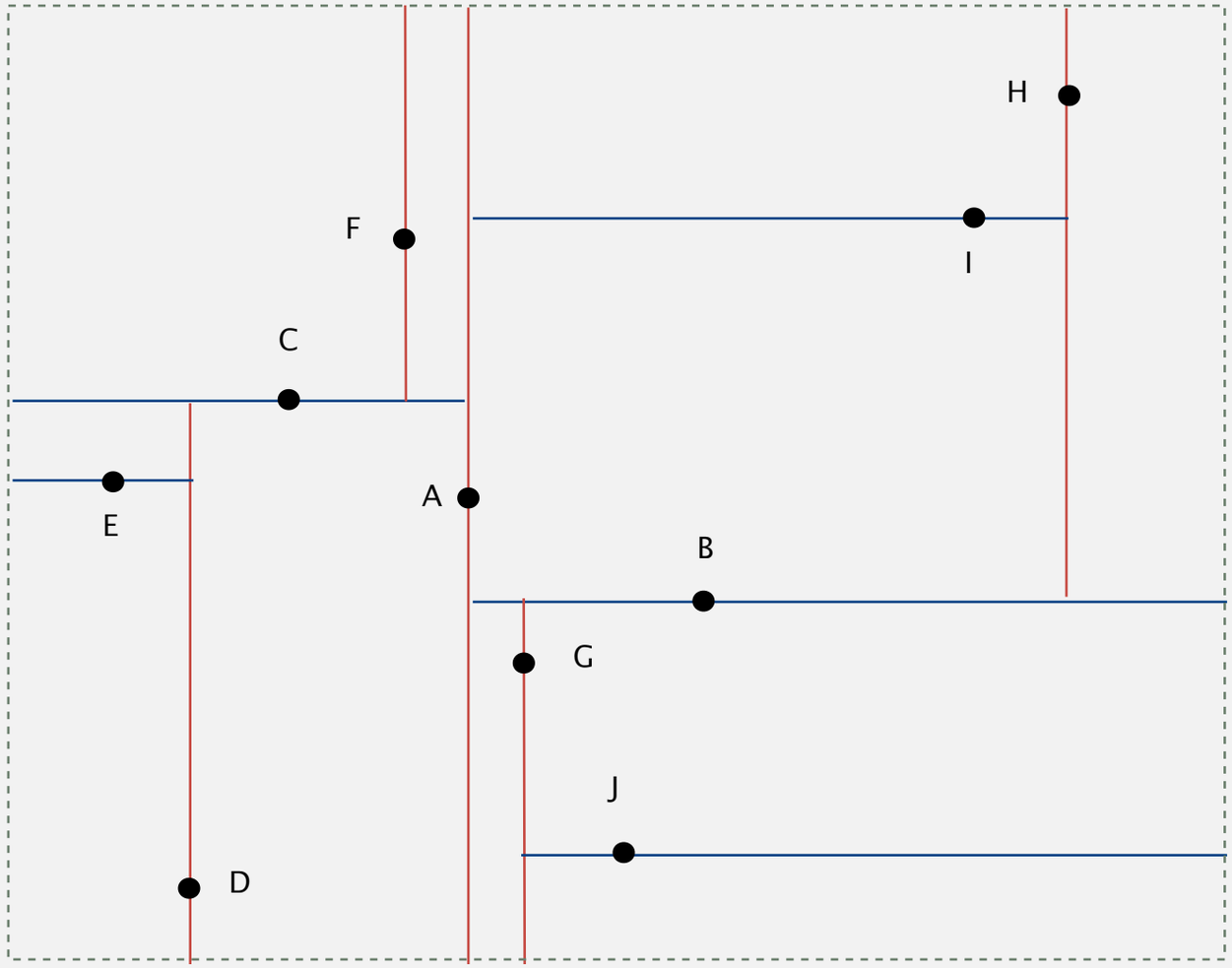
Recursively partition plane into two halfplanes.

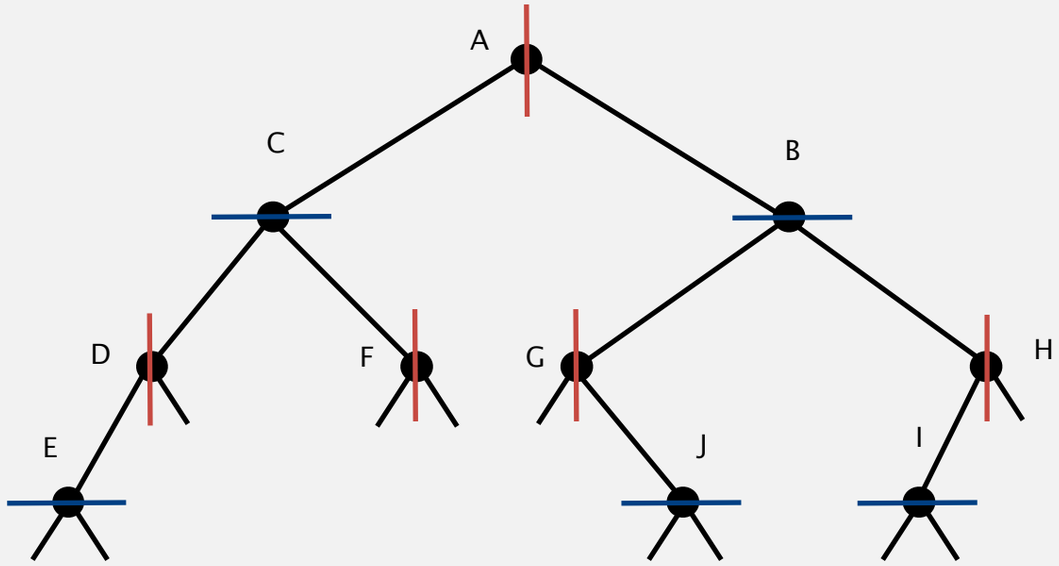




2d tree demo: insertion

Recursively partition plane into two halfplanes.





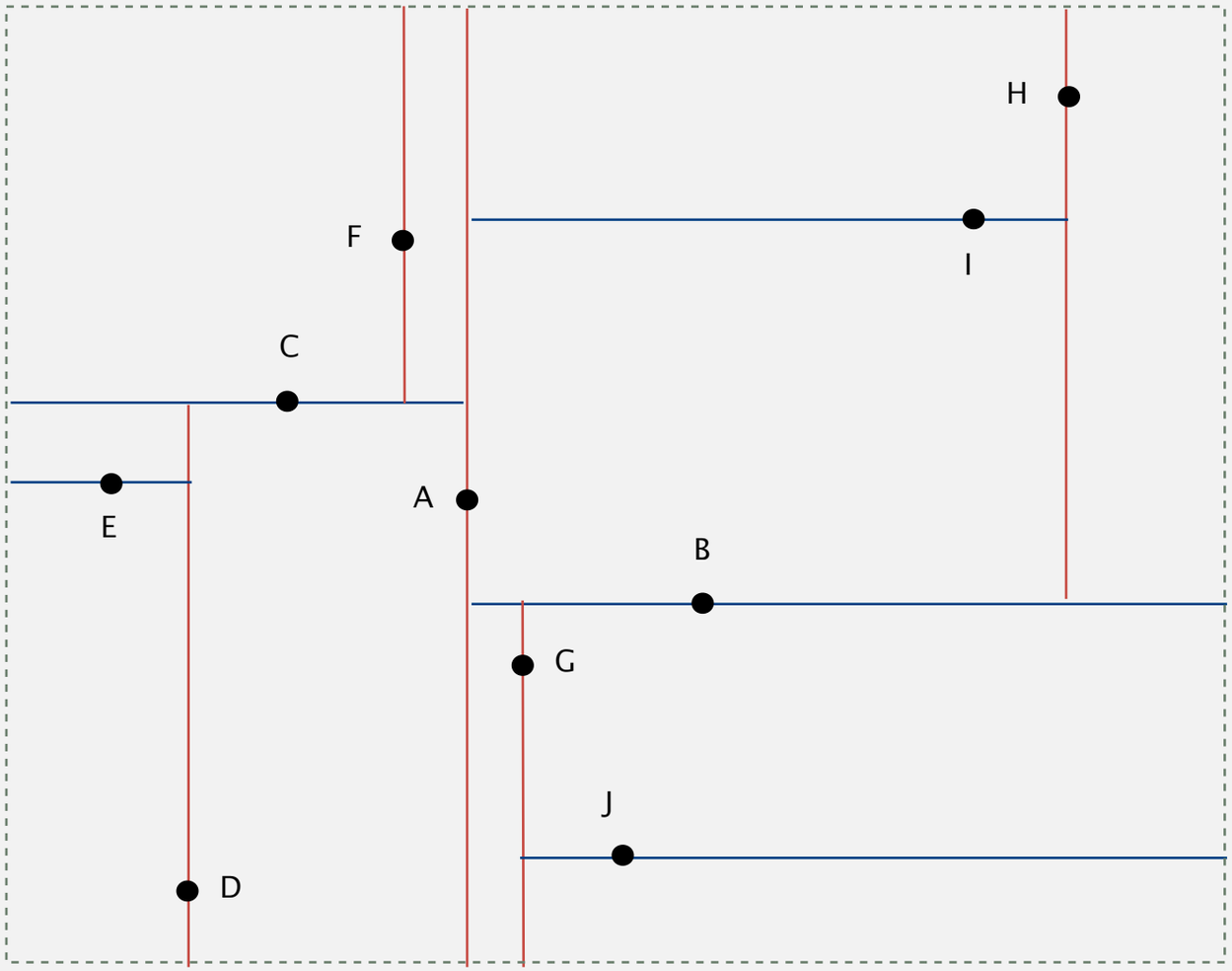
https://algs4.cs.princeton.edu

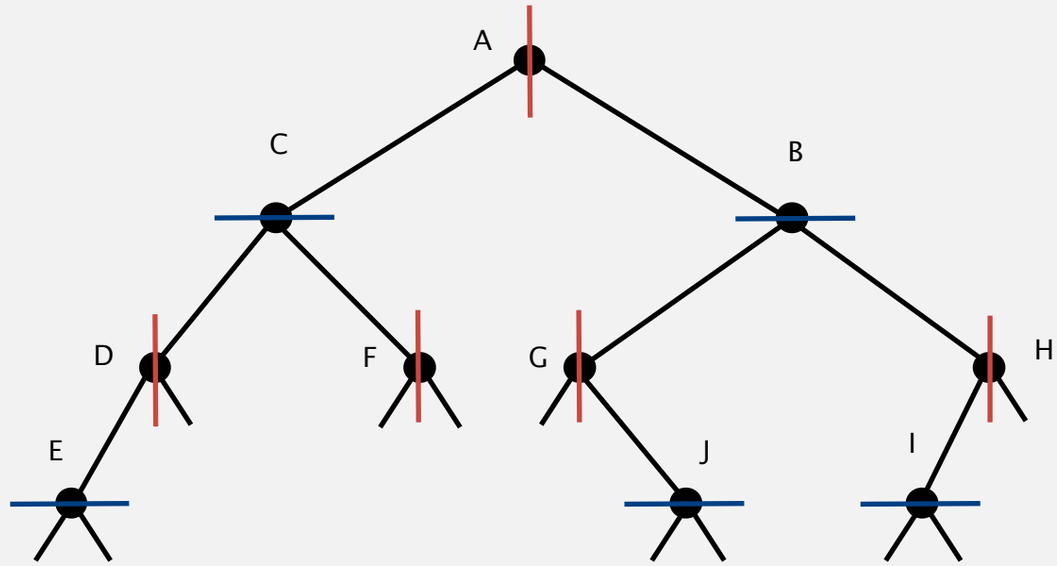
2d Tree Demo

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

2d tree demo: range search

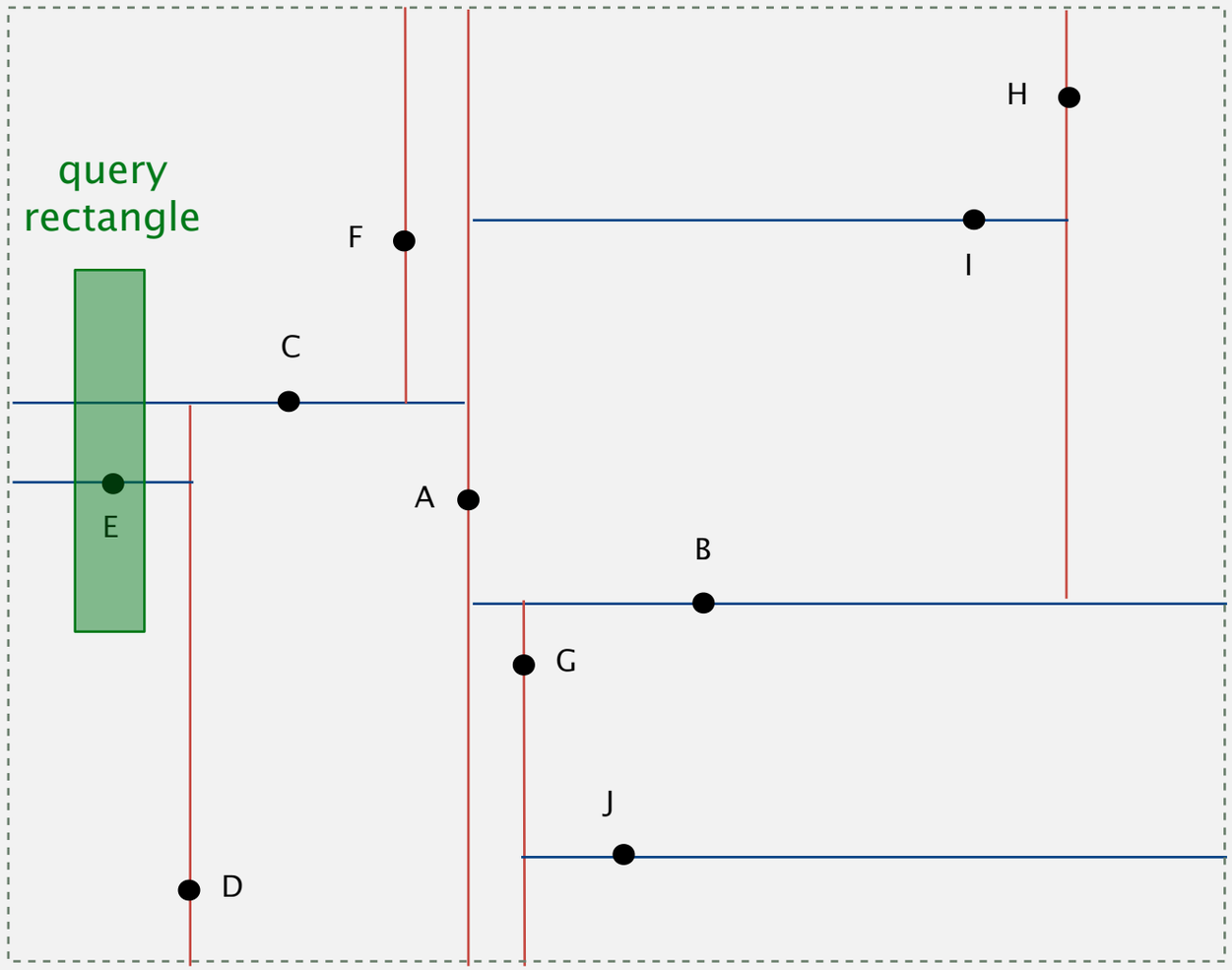
Goal. Find all points in a query rectangle.

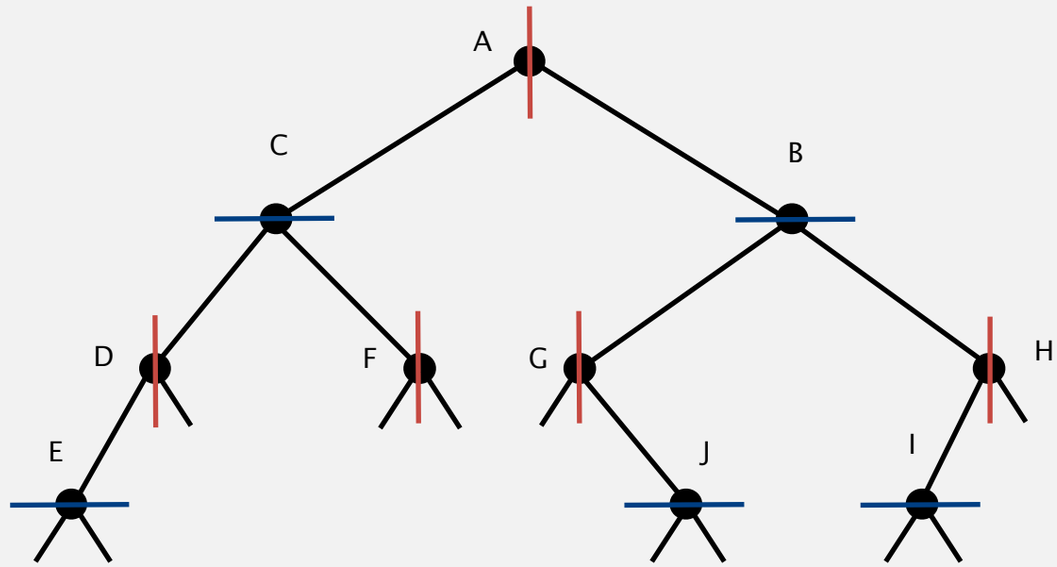




2d tree demo: range search

Goal. Find all points in a query rectangle.



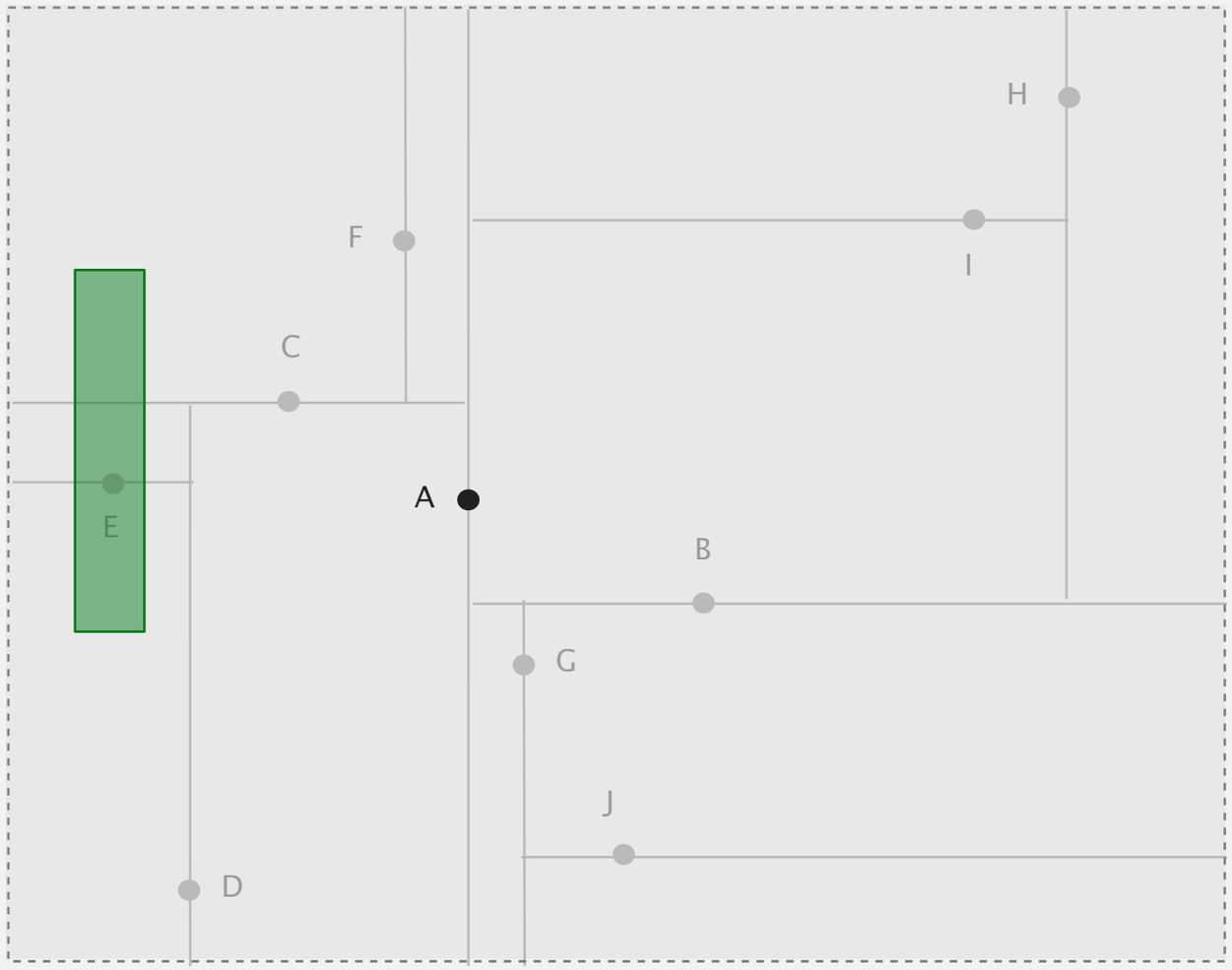


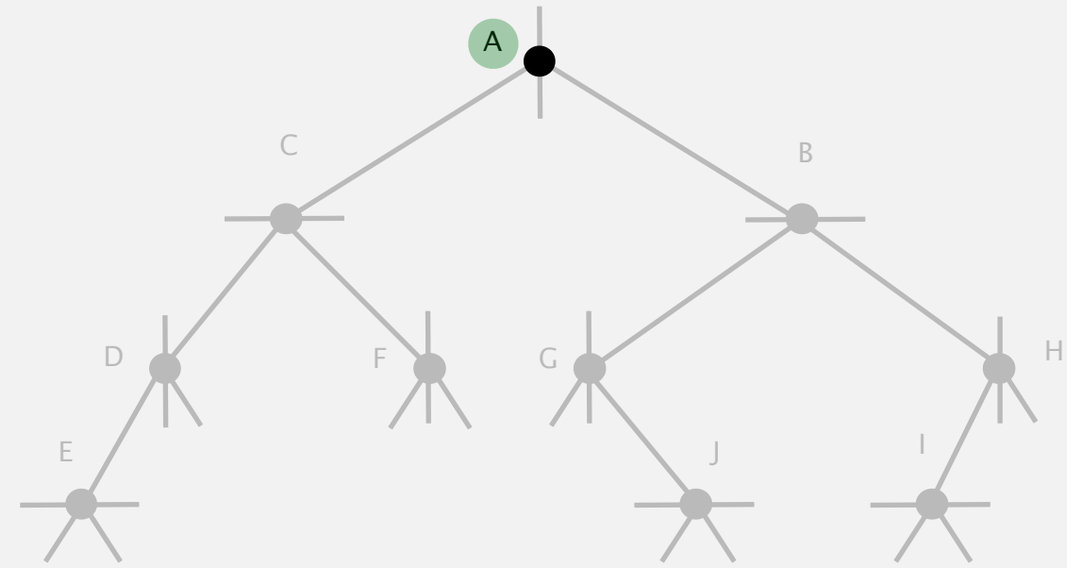
(xmin, ymin, xmax, ymax) = (0.053, 0.393, 0.106, 0.683)

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.





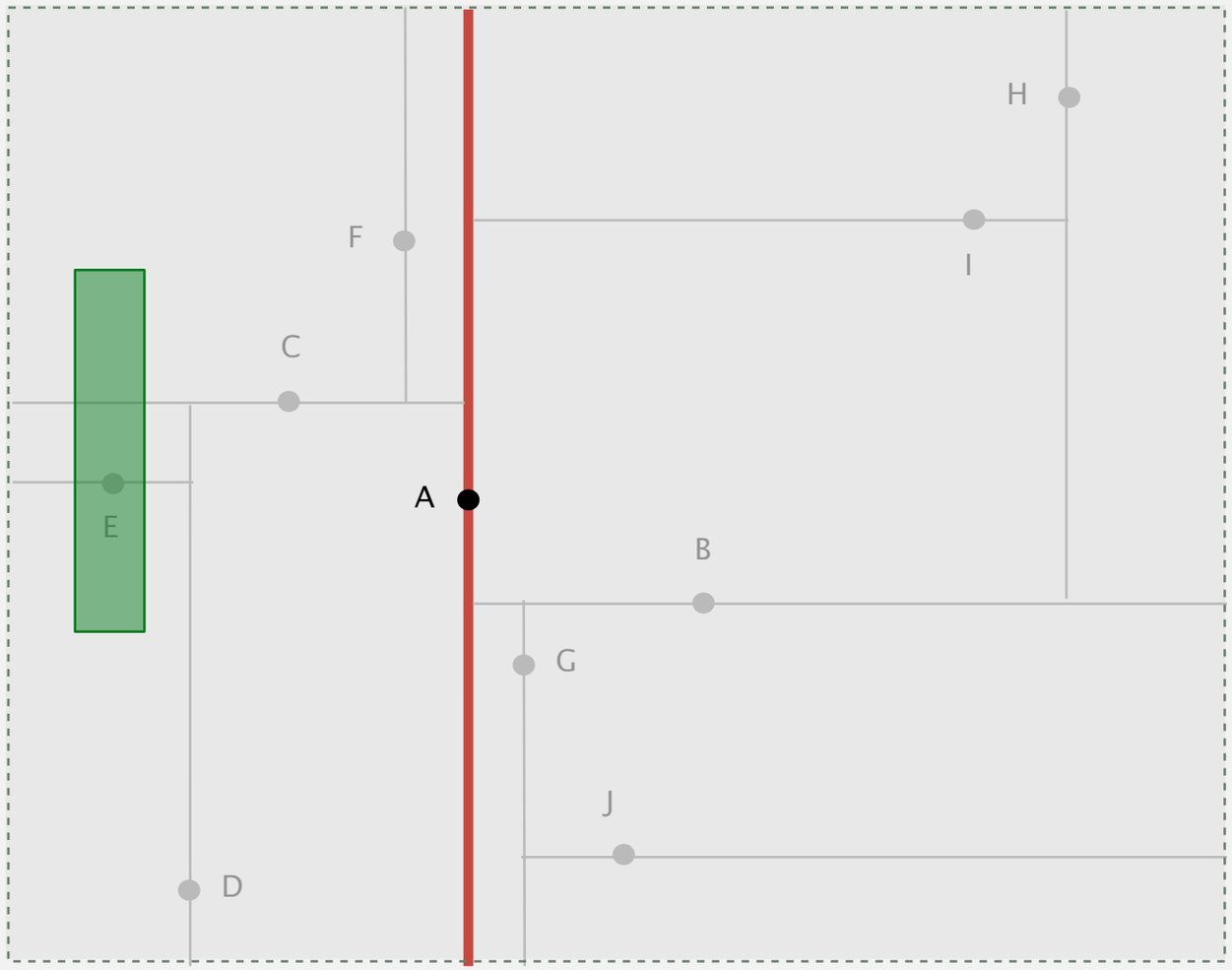
search root node

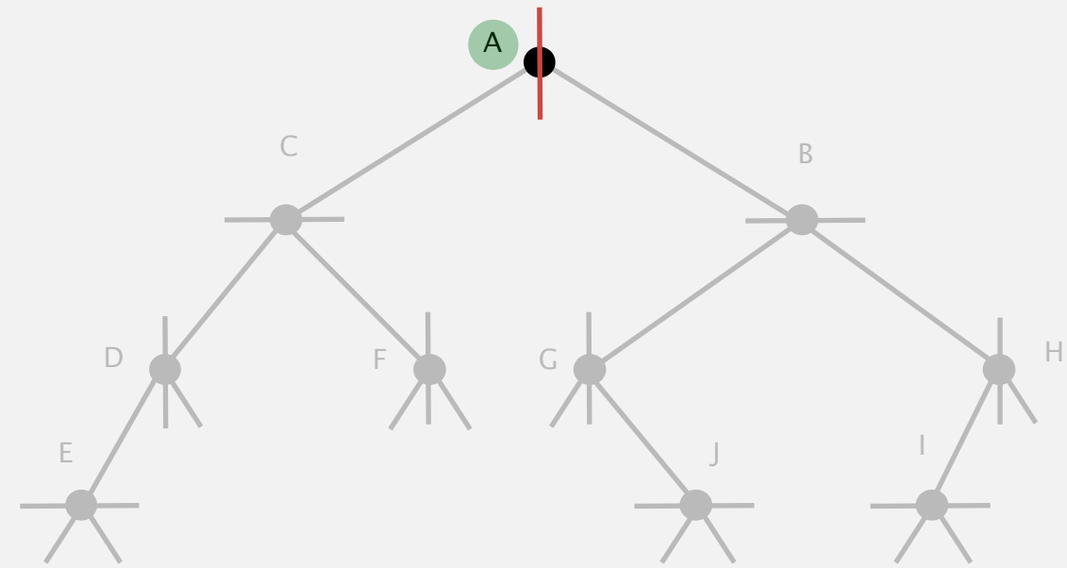
check if query rectangle contains point A

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





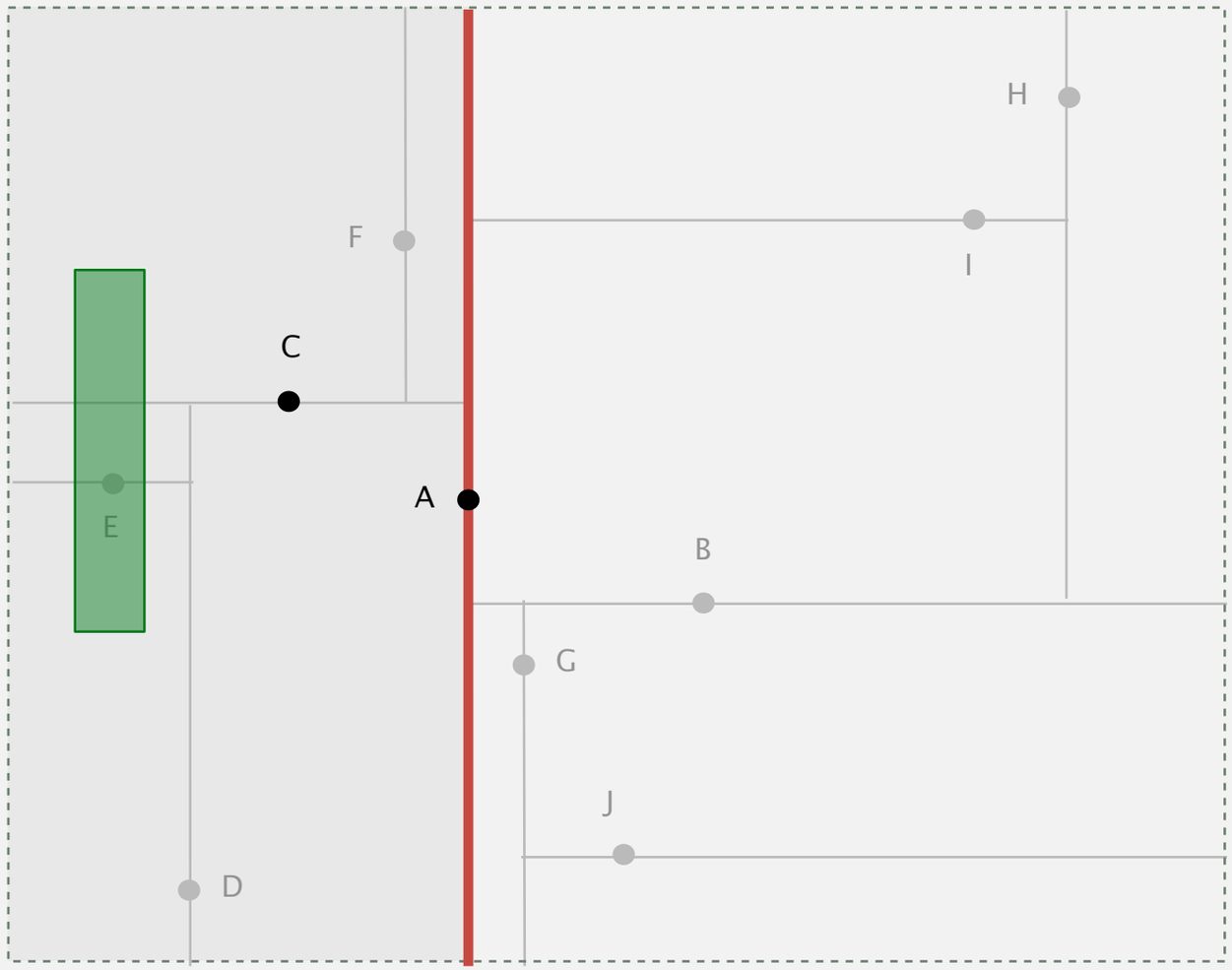
query rectangle to left of splitting line

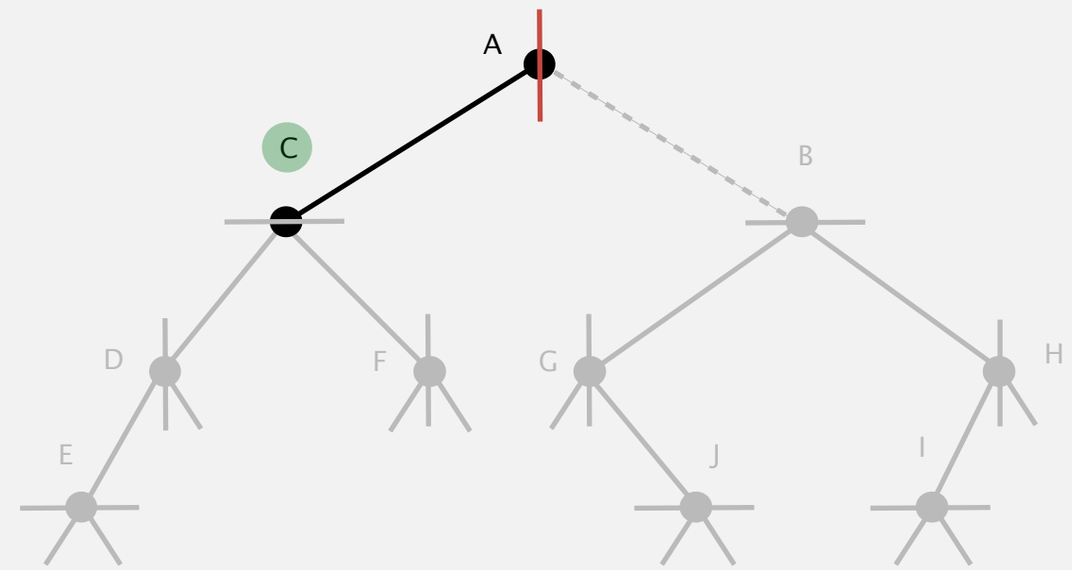
search only in left subtree

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





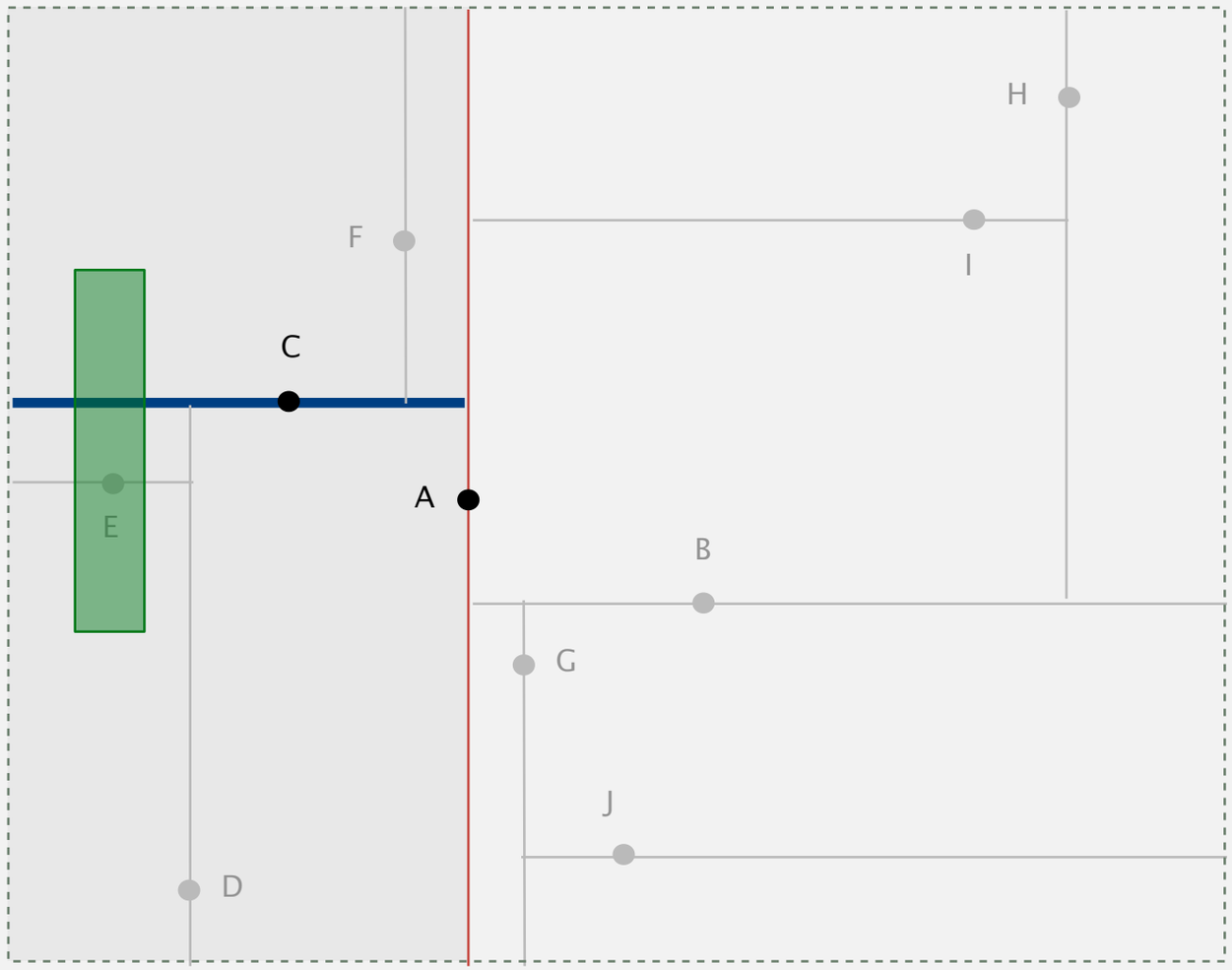
search left subtree

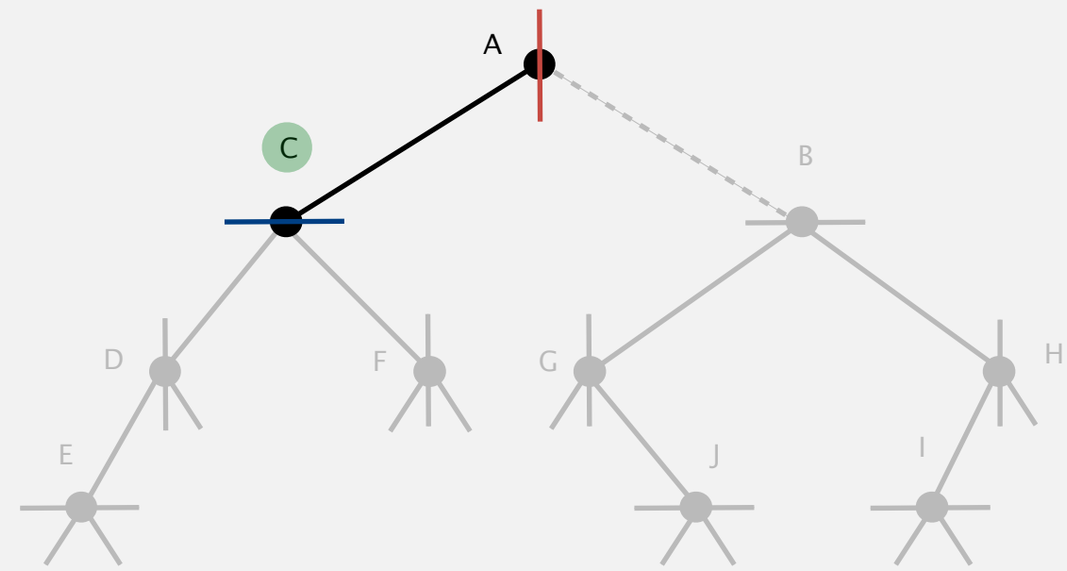
check if query rectangle contains point C

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





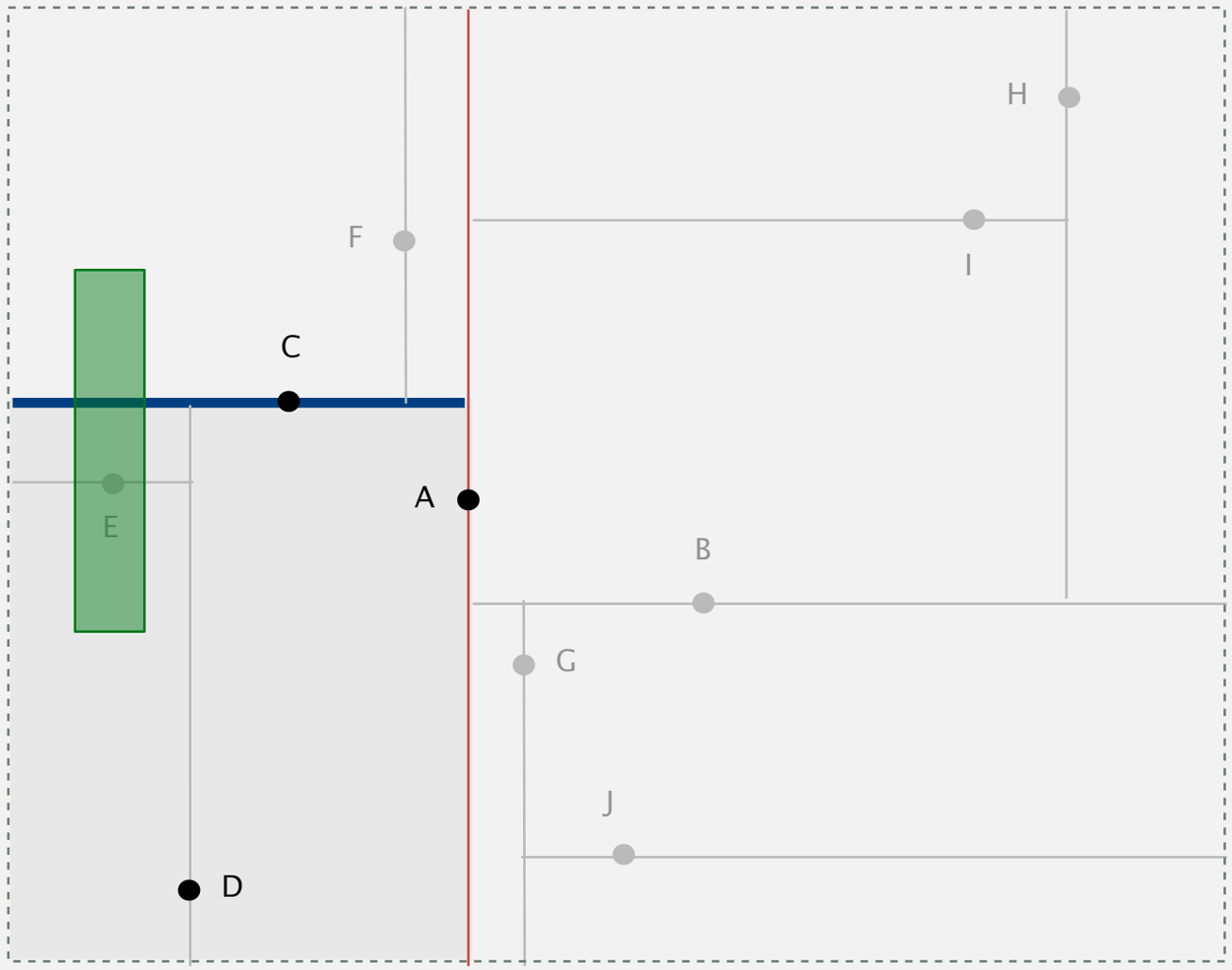
query rectangle intersects splitting line

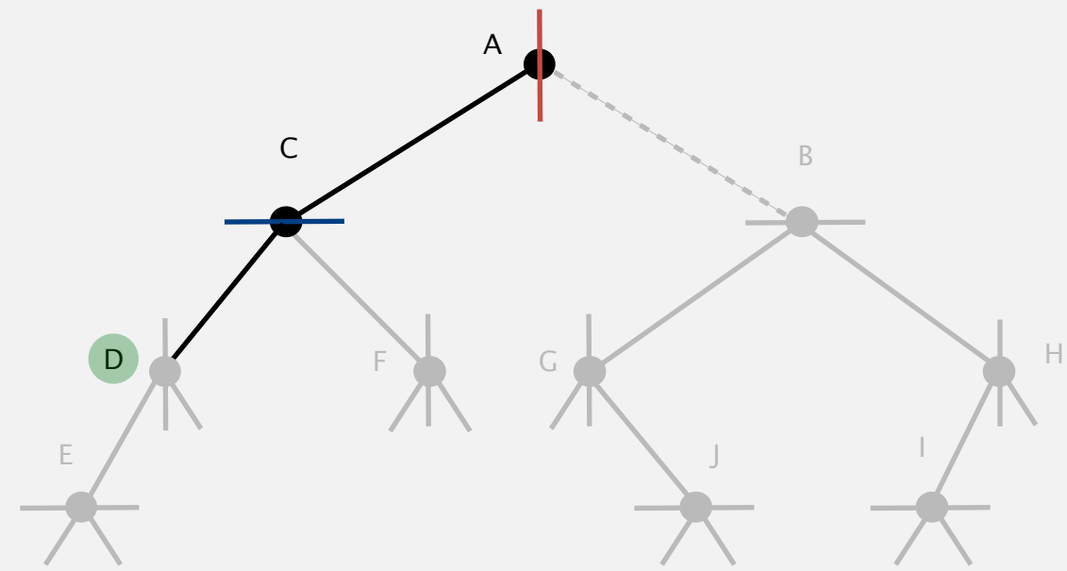
search bottom and top subtrees

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





search left subtree

check if query rectangle contains point D

Geometric applications of BSTs: quiz 3

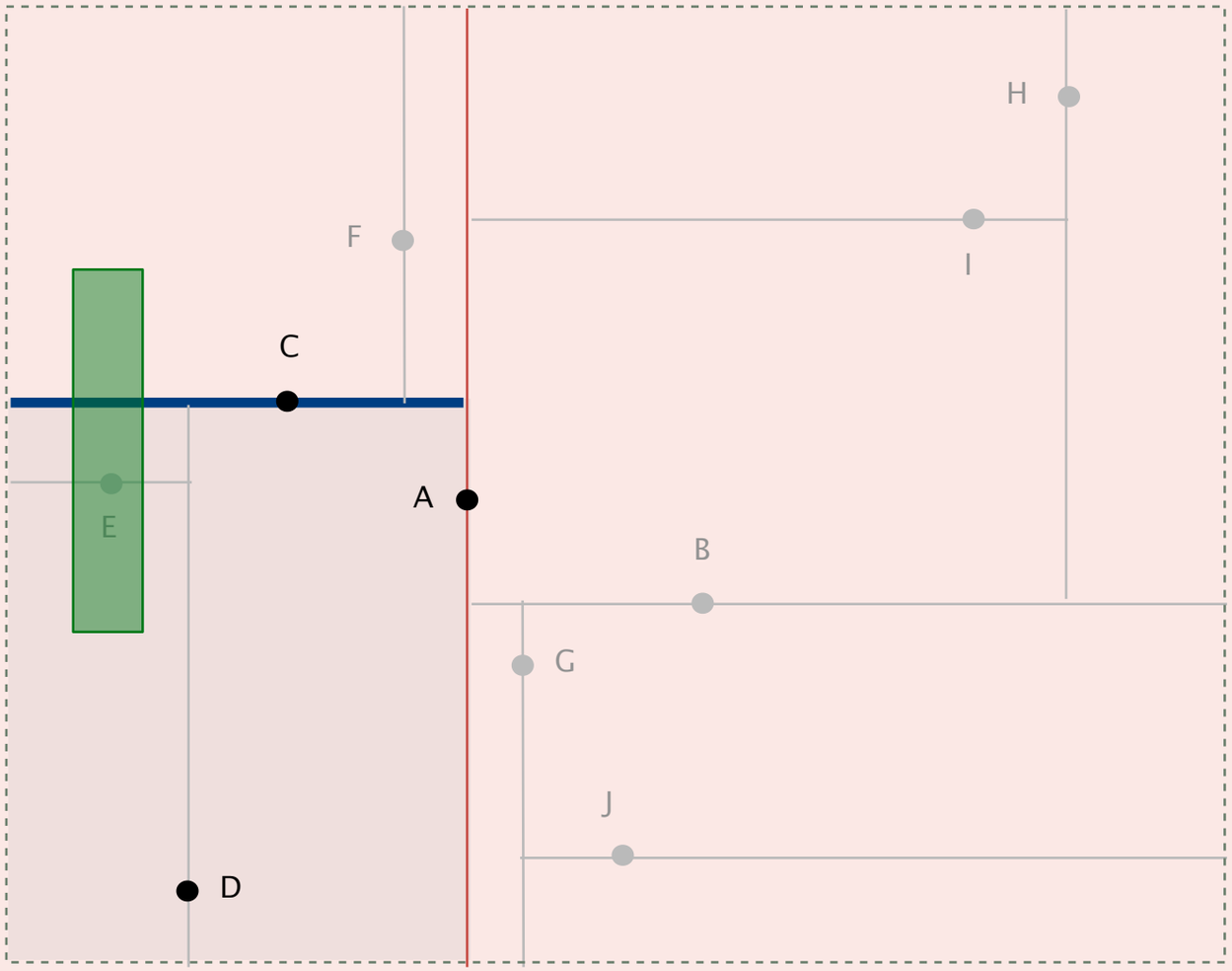
Which subtrees of D should we explore?

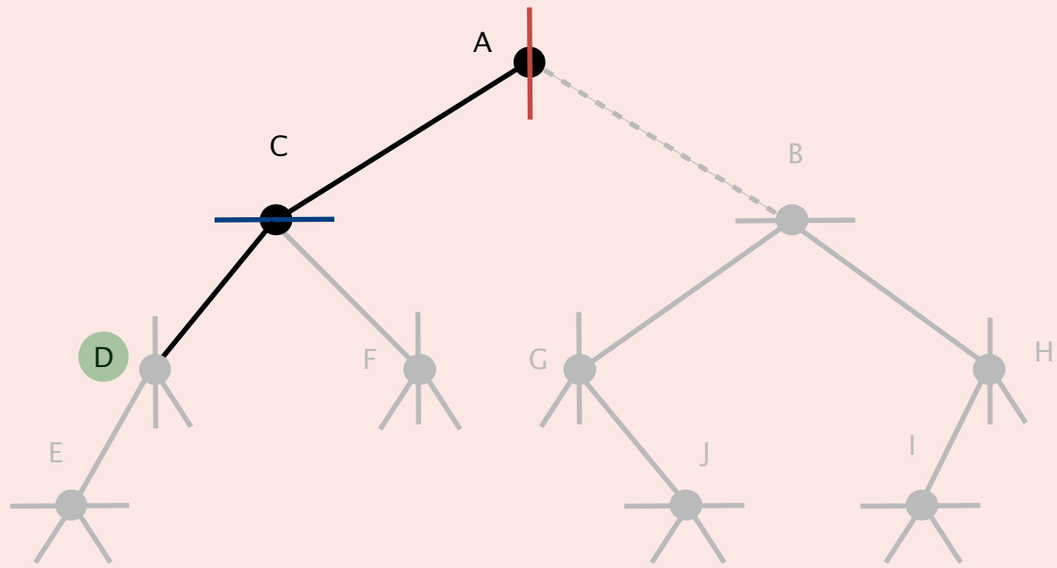
A. Left subtree only.

B. Right subtree only.

C. Both left and right subtrees.

D. Neither left nor right subtrees.

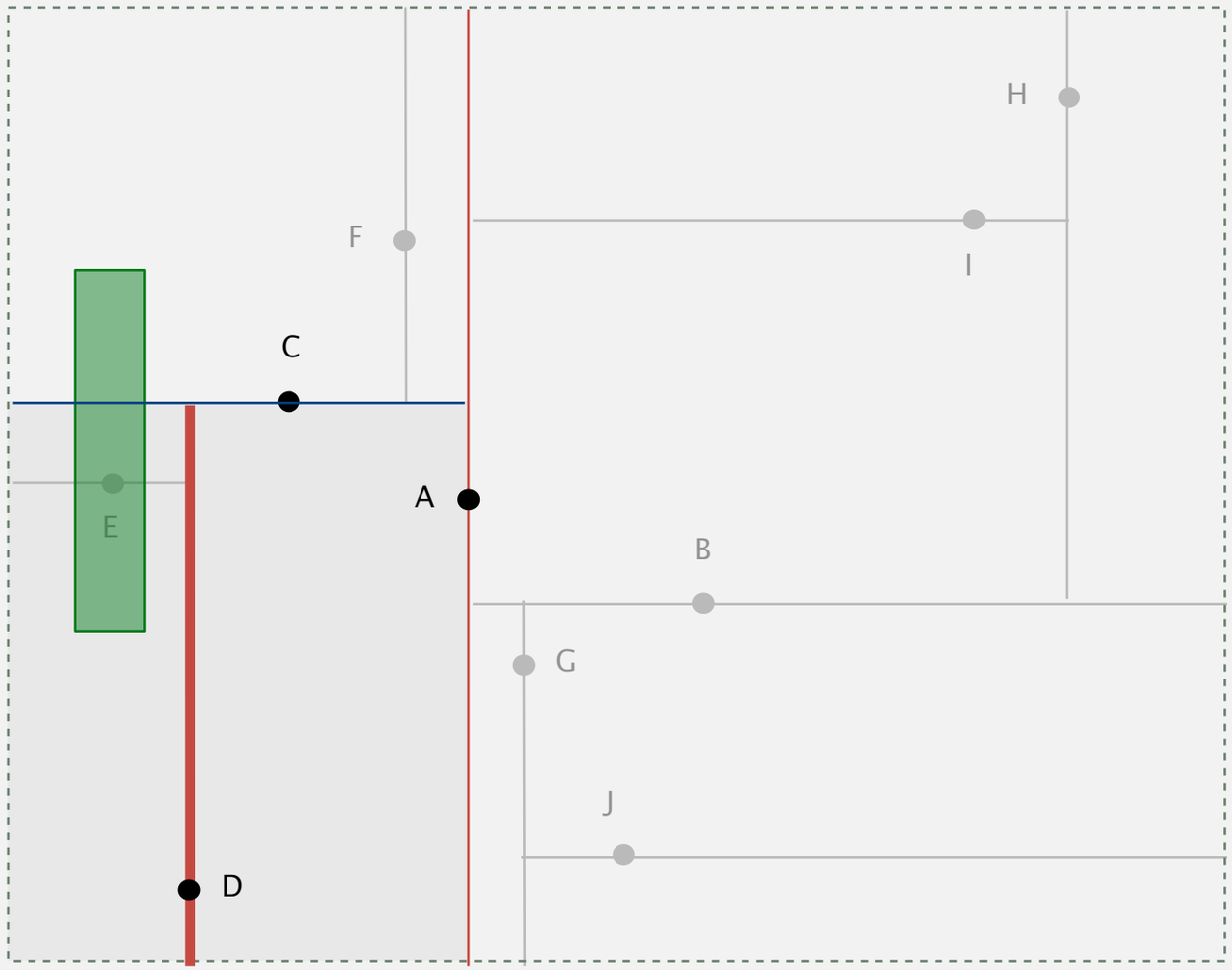


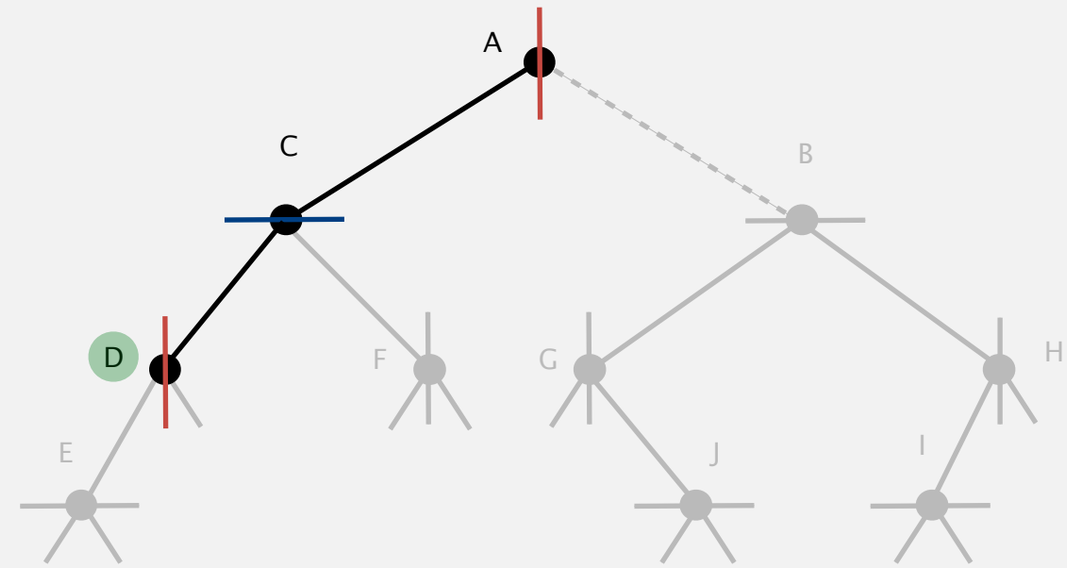


2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





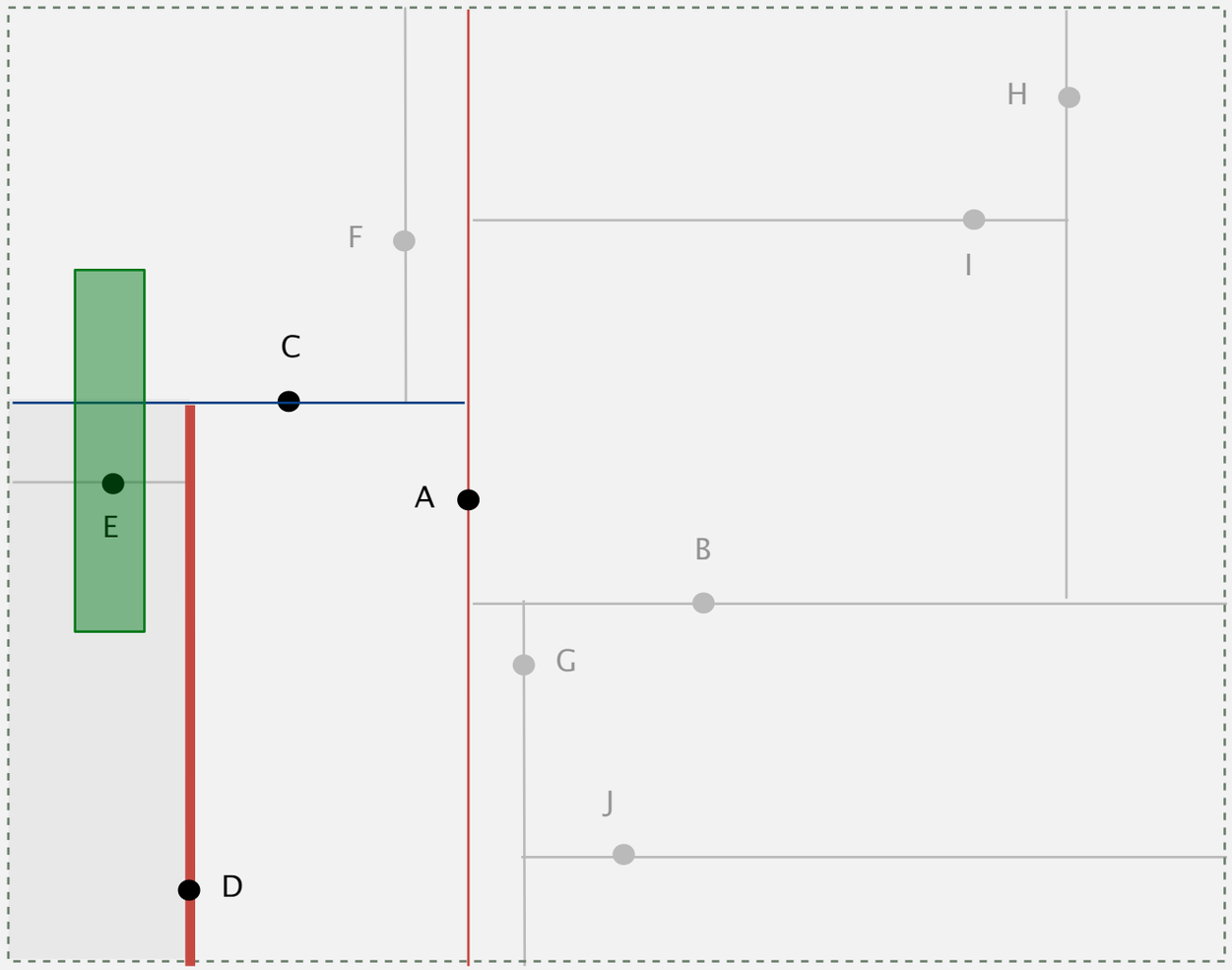
query rectangle to left of splitting line

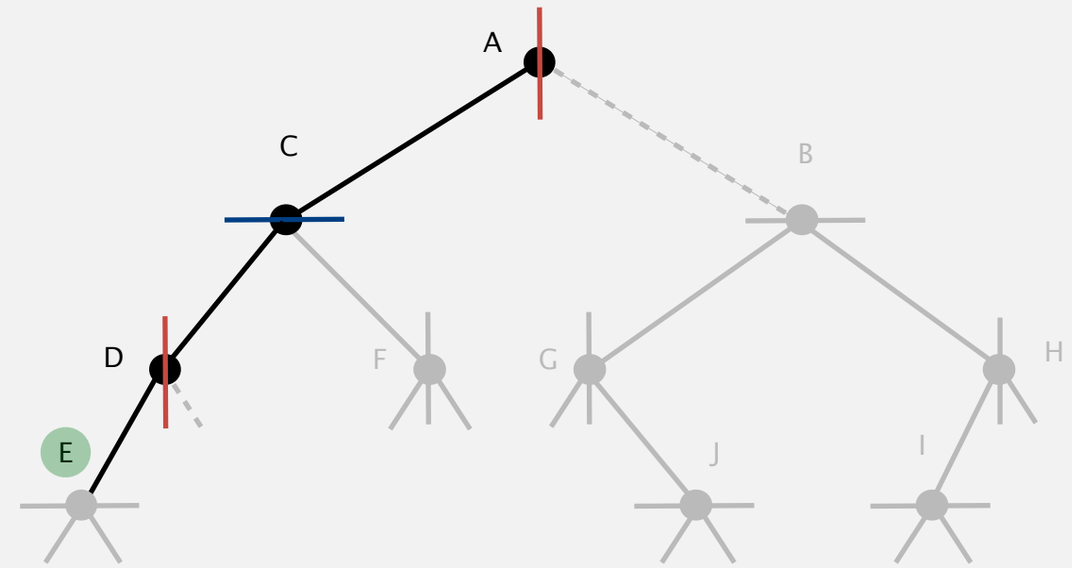
search only in left subtree

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





search left subtree

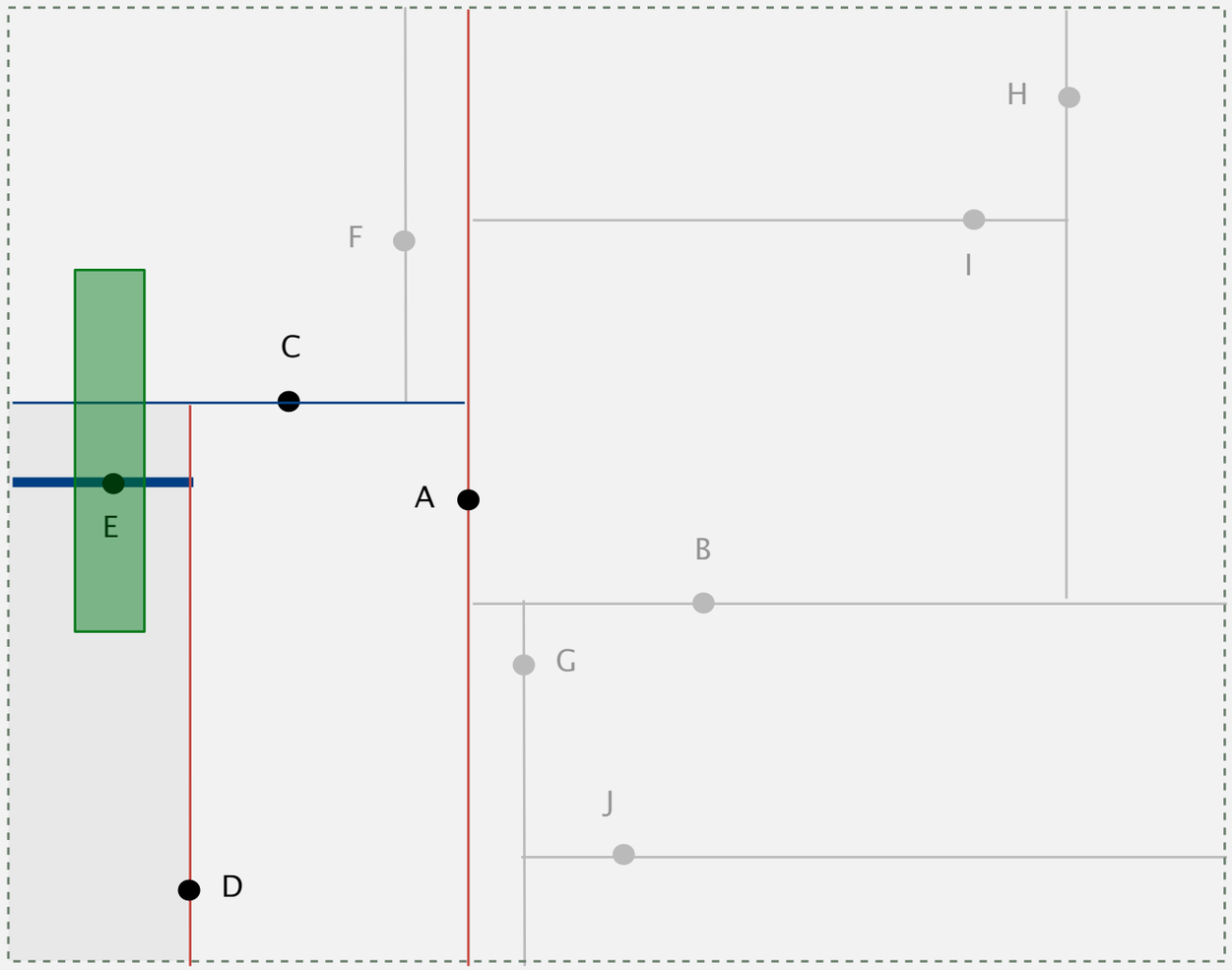
check if query rectangle contains point E

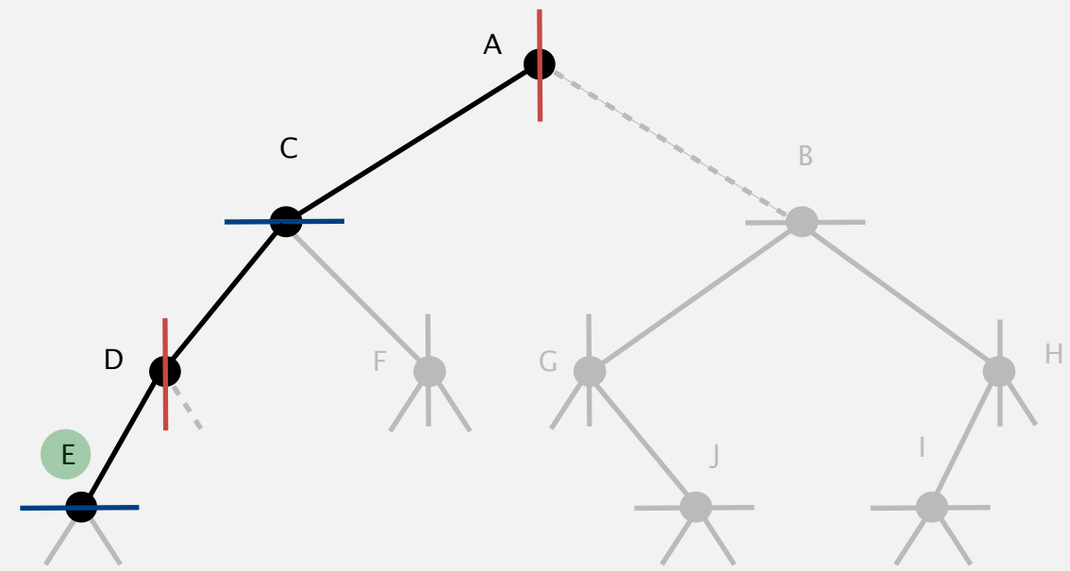
(search hit)

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





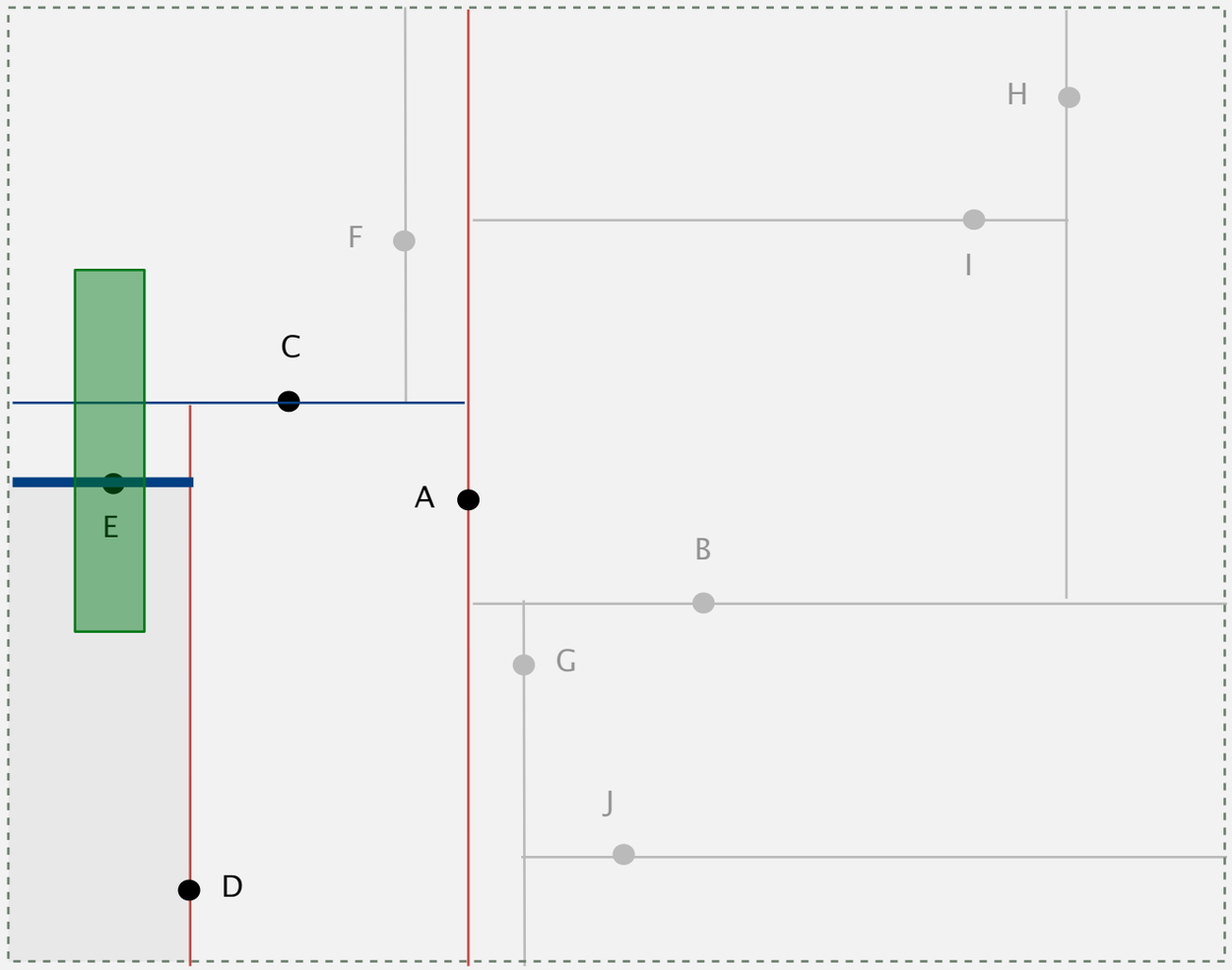
query rectangle intersects splitting line

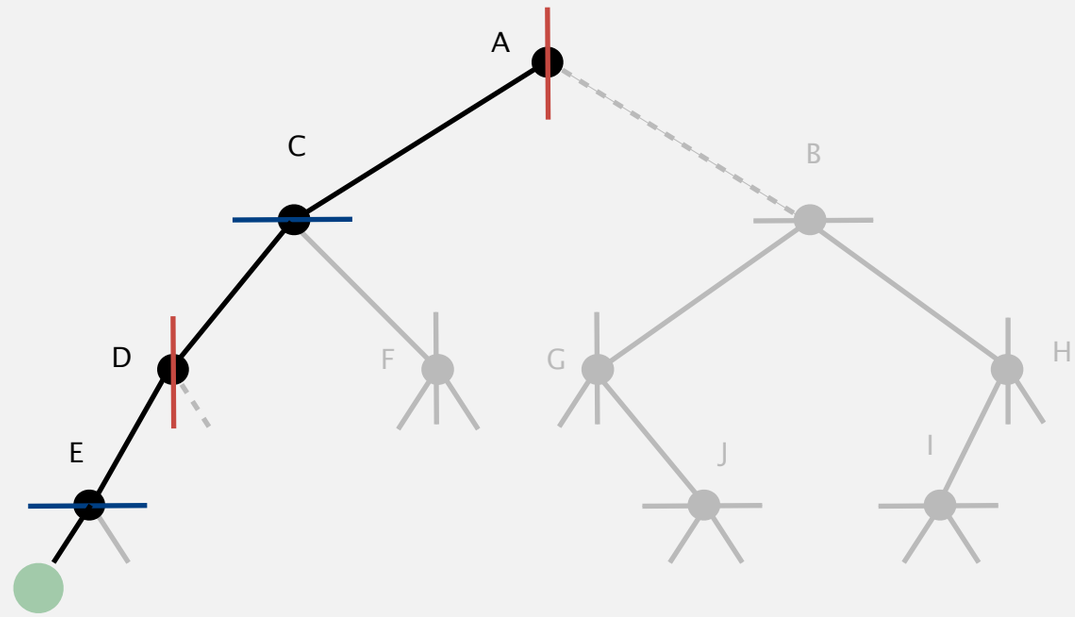
search bottom and top subtrees

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





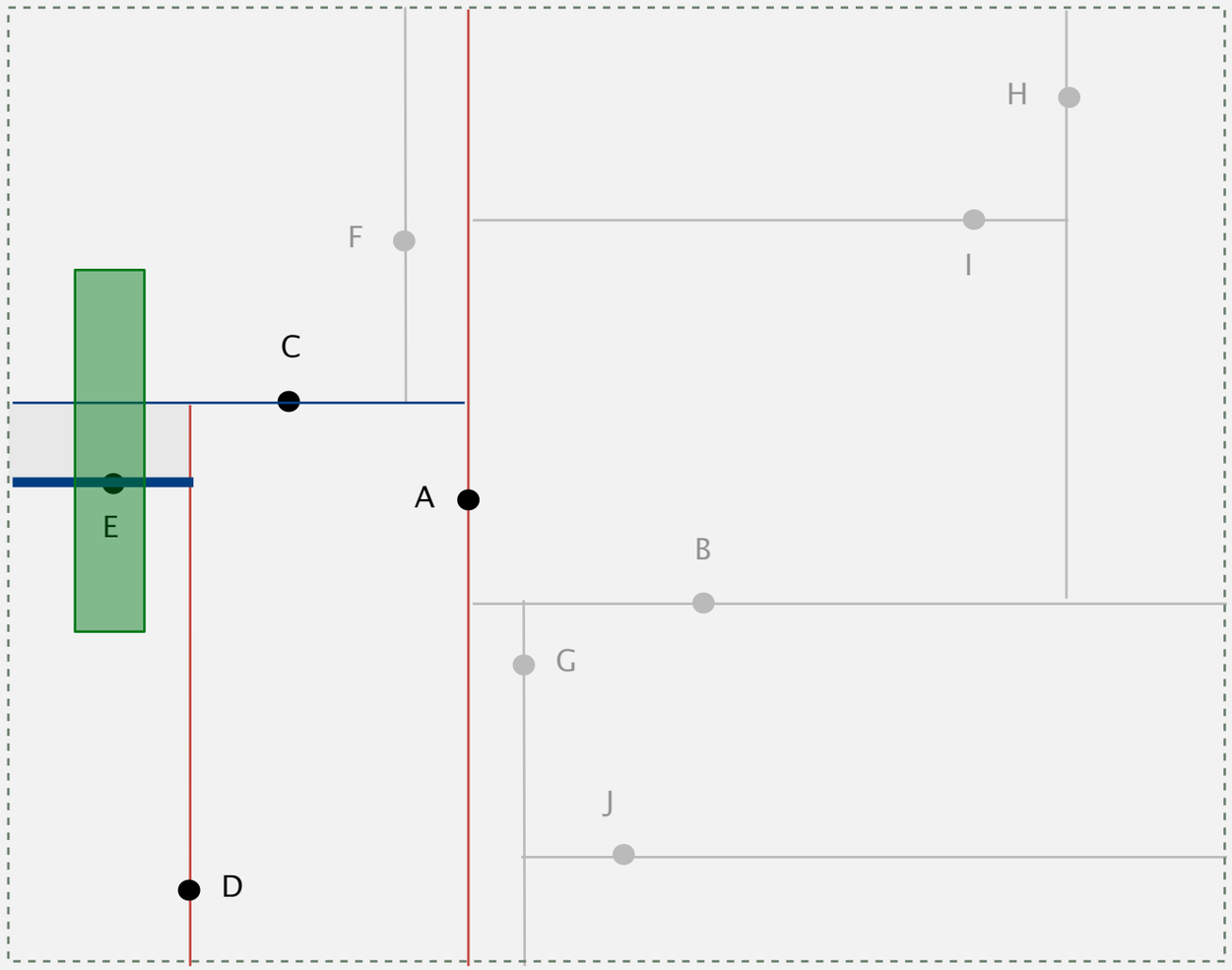
search bottom subtree

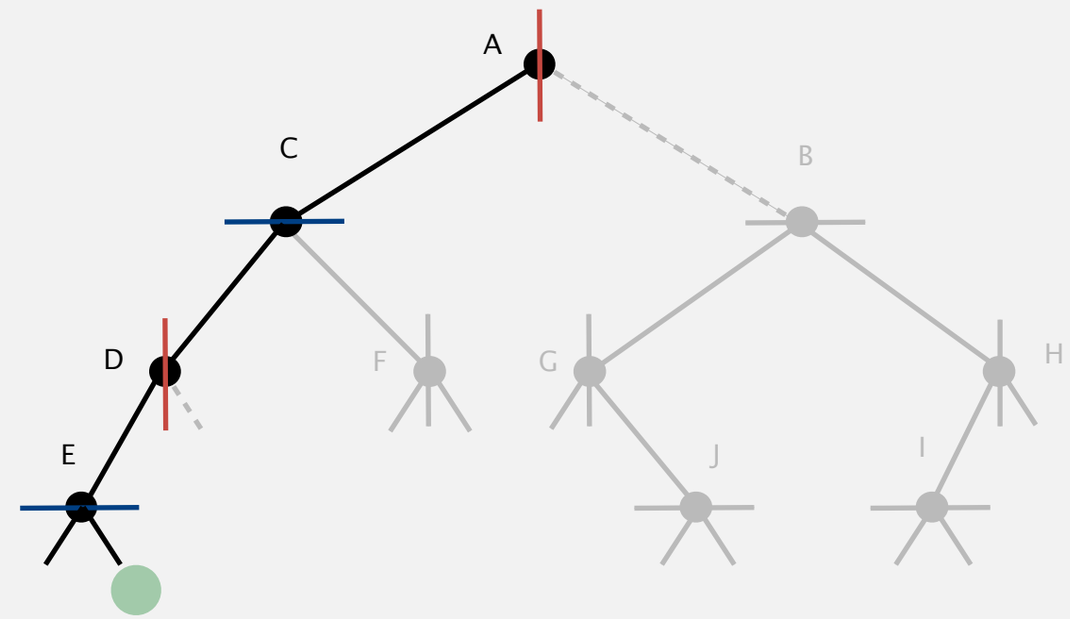
stop since empty

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





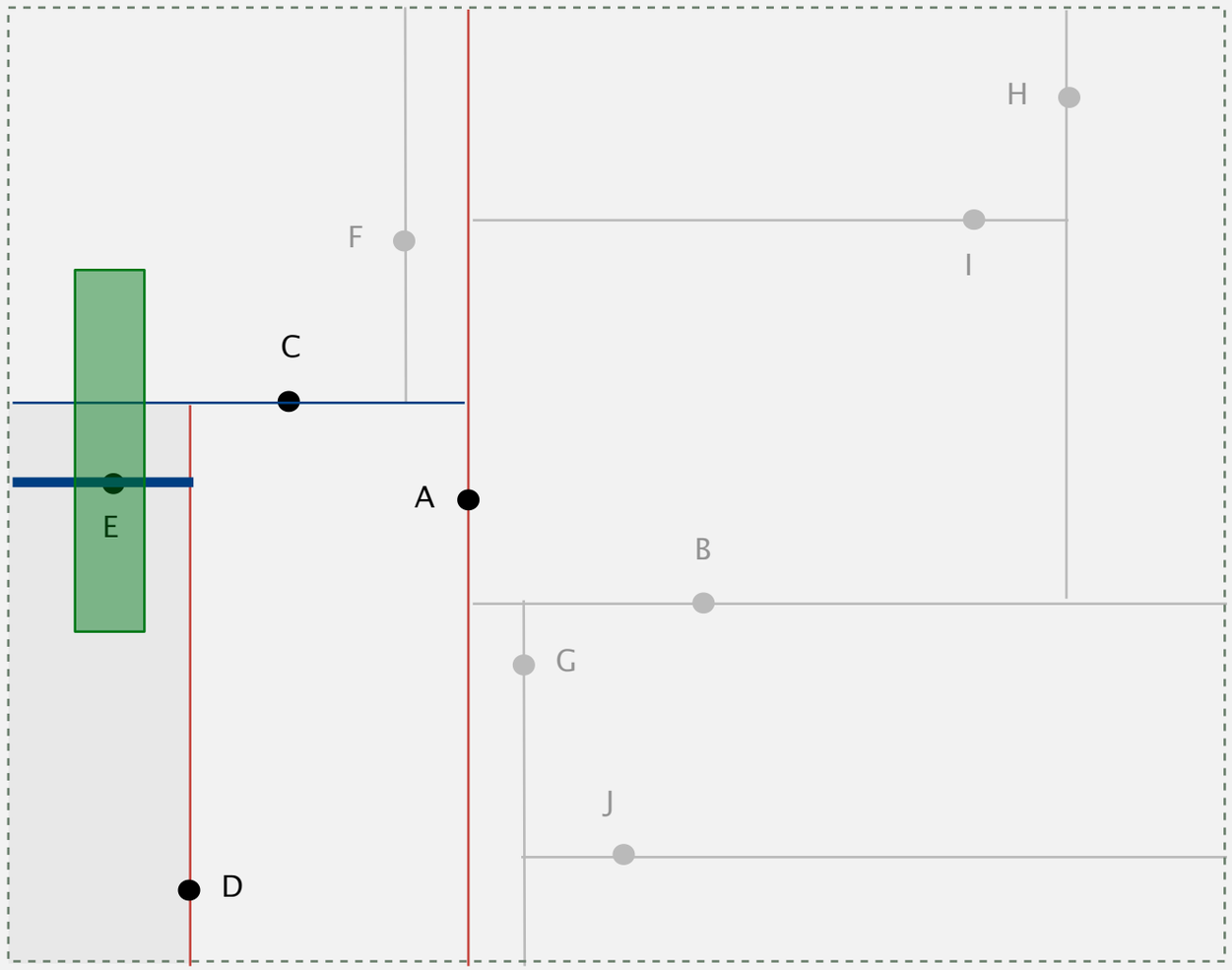
search top subtree

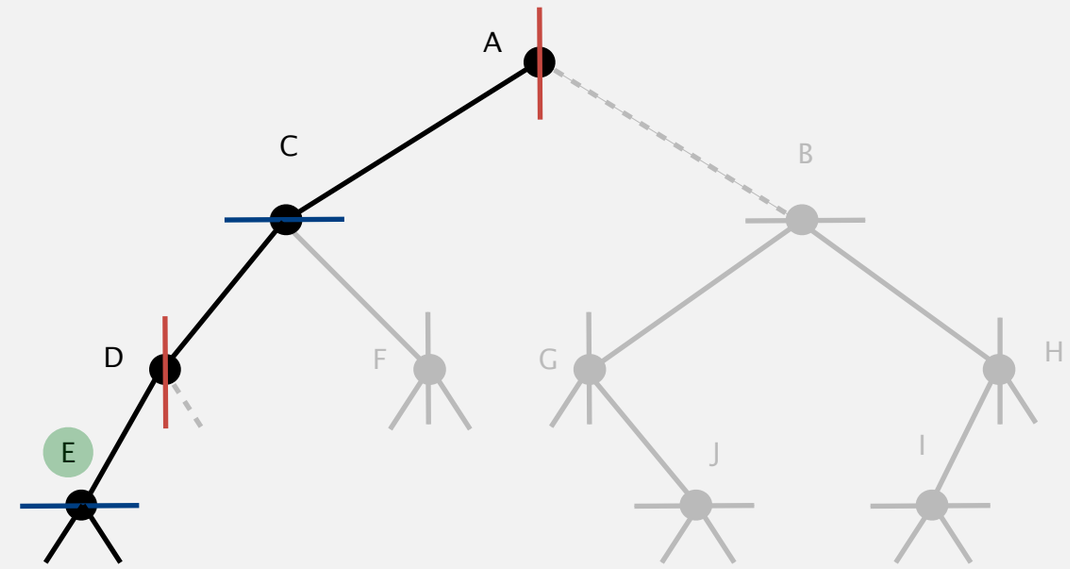
stop since empty

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.



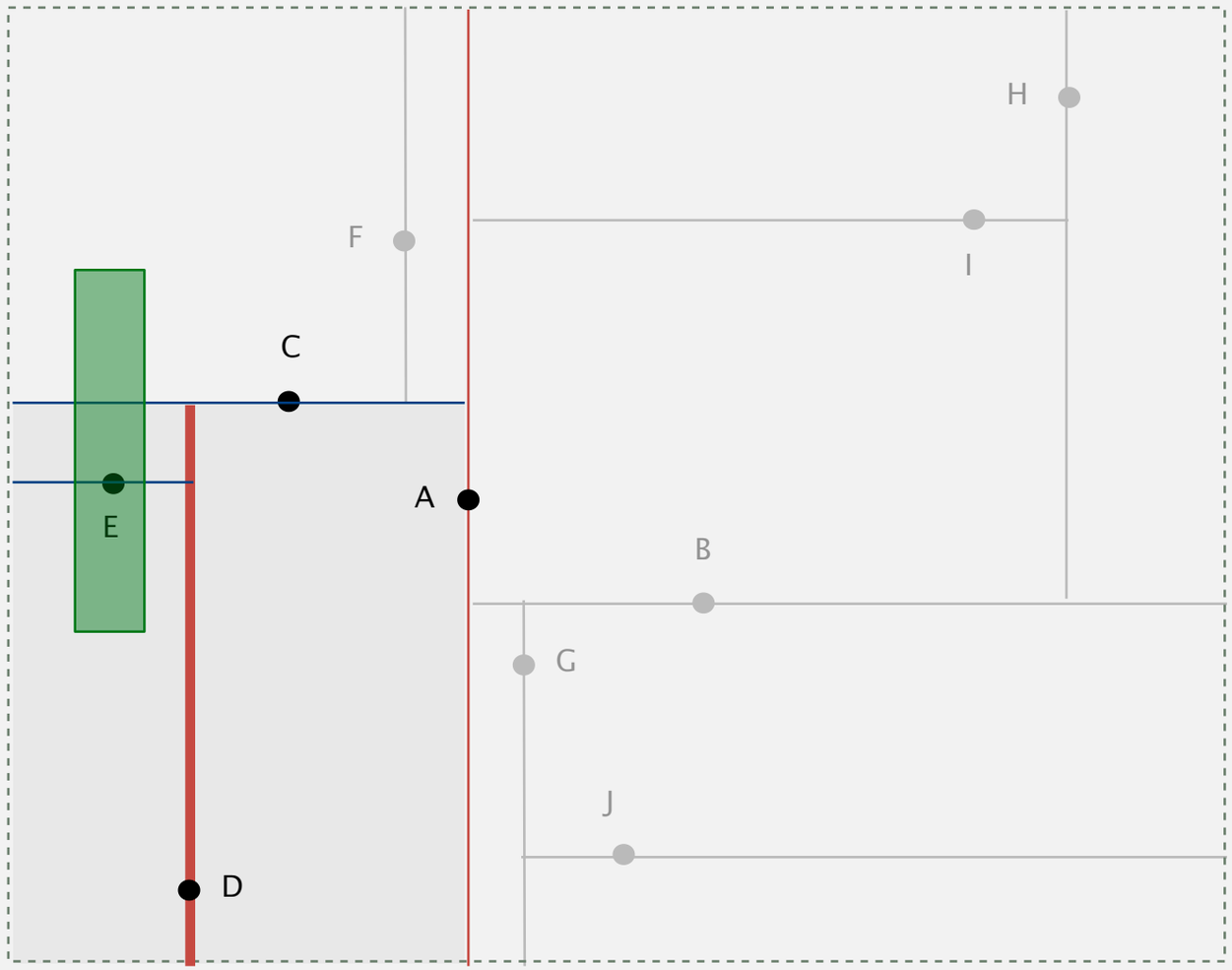


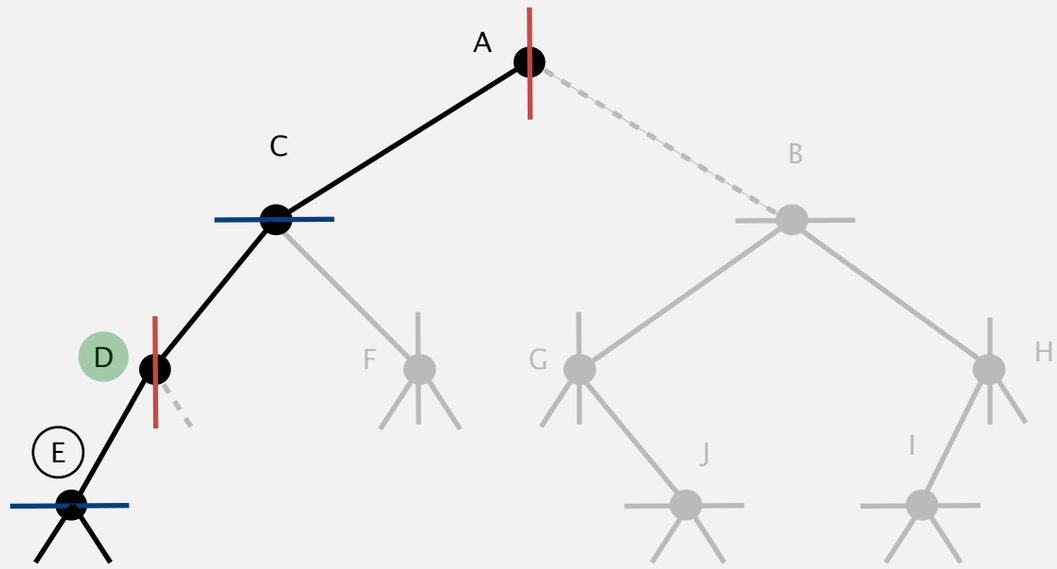
return from function call

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.



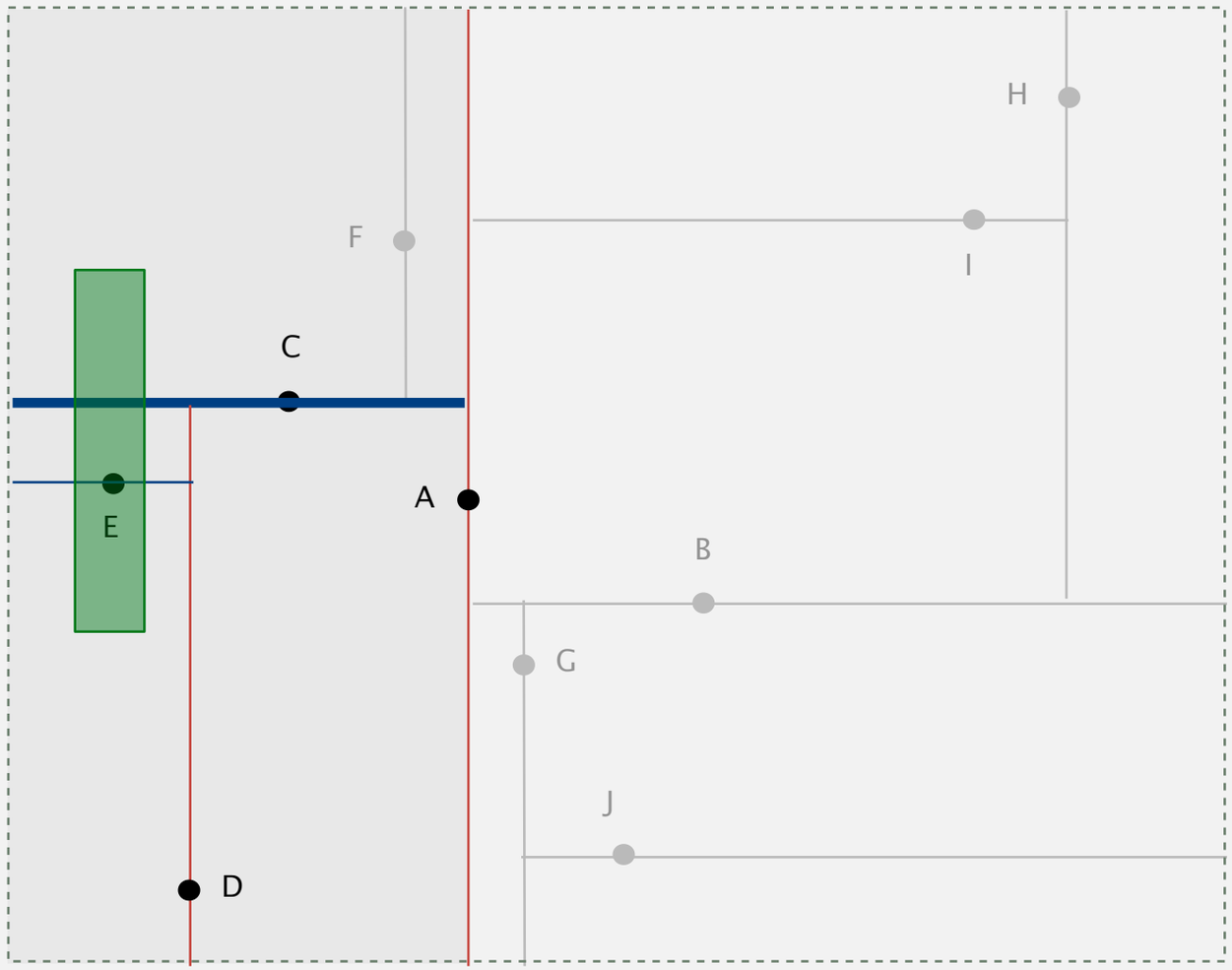


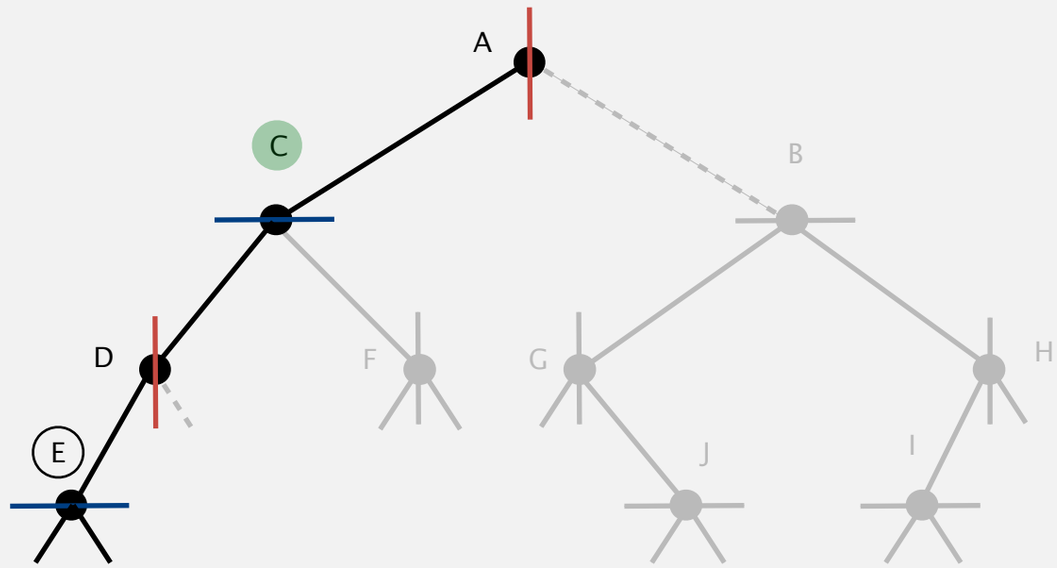
return from function call

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.



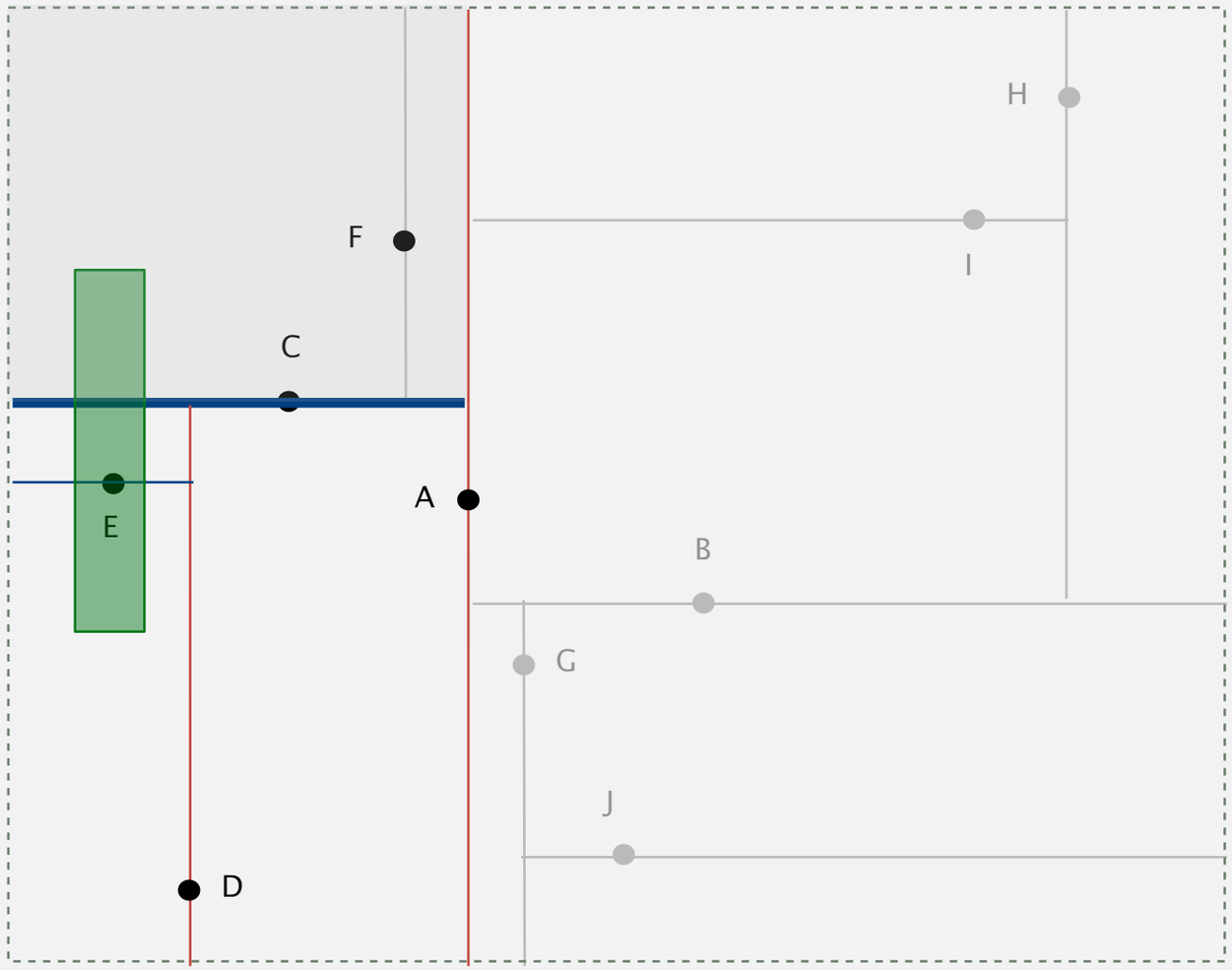


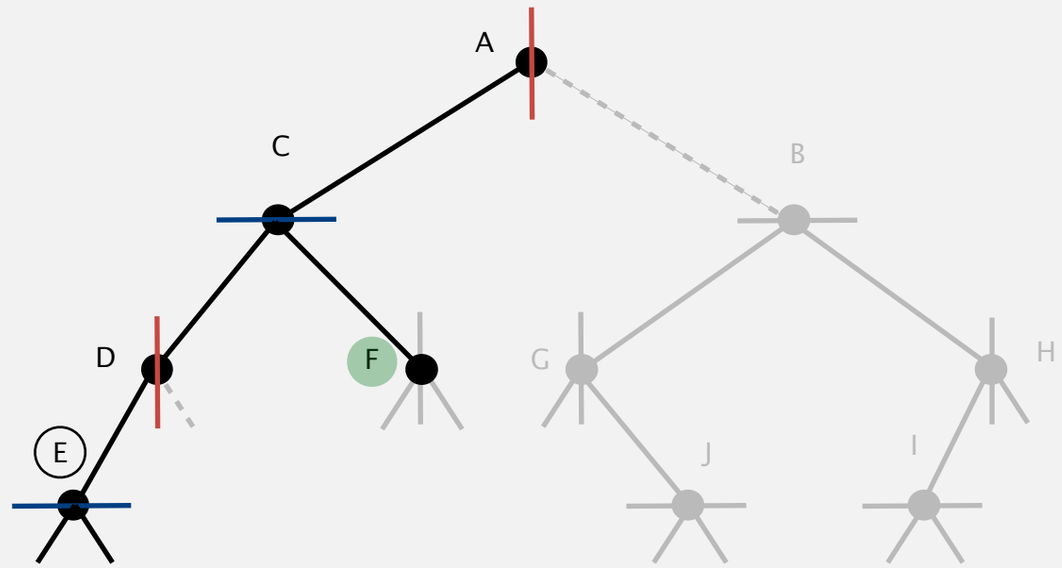
return from function call

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





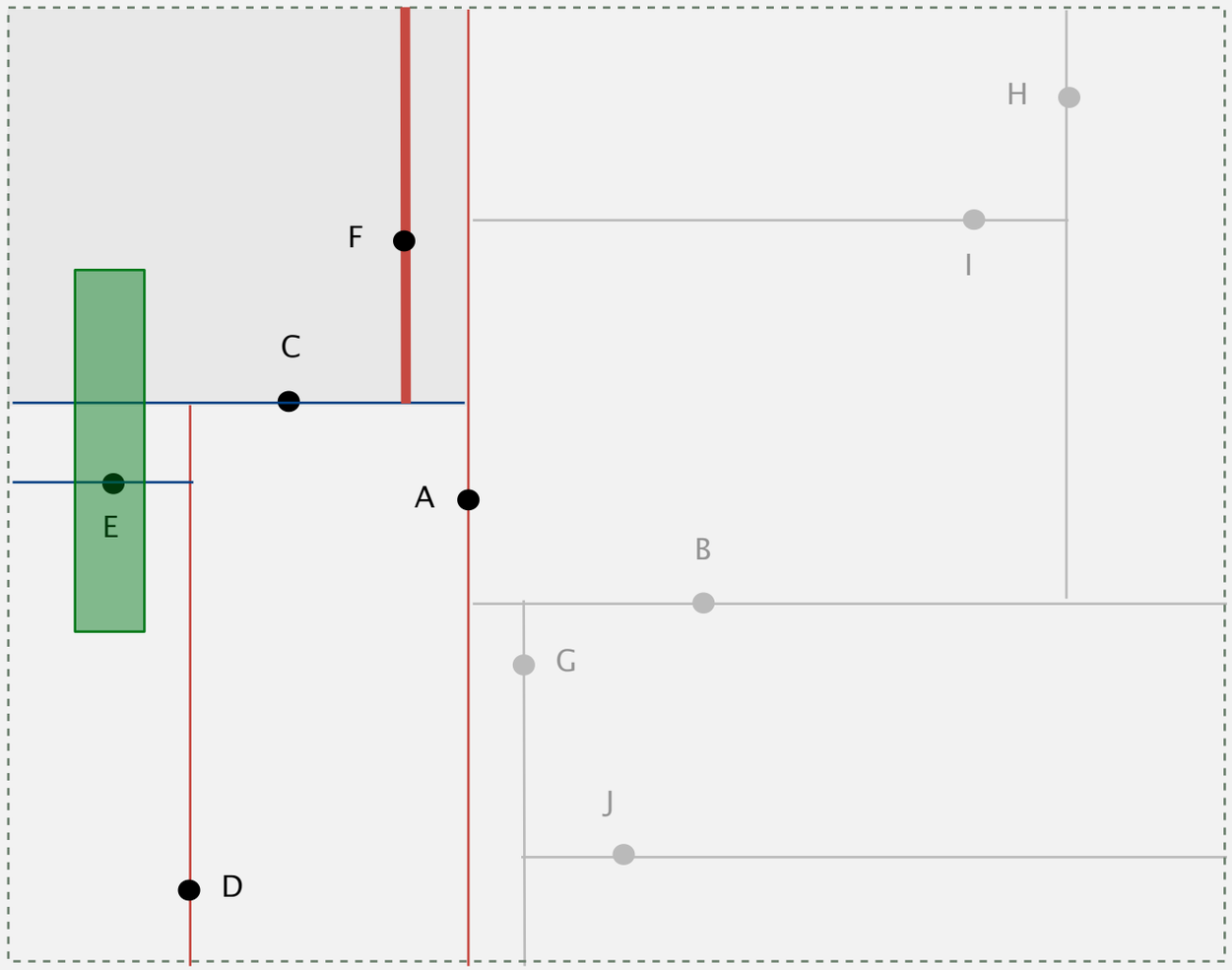
search top subtree

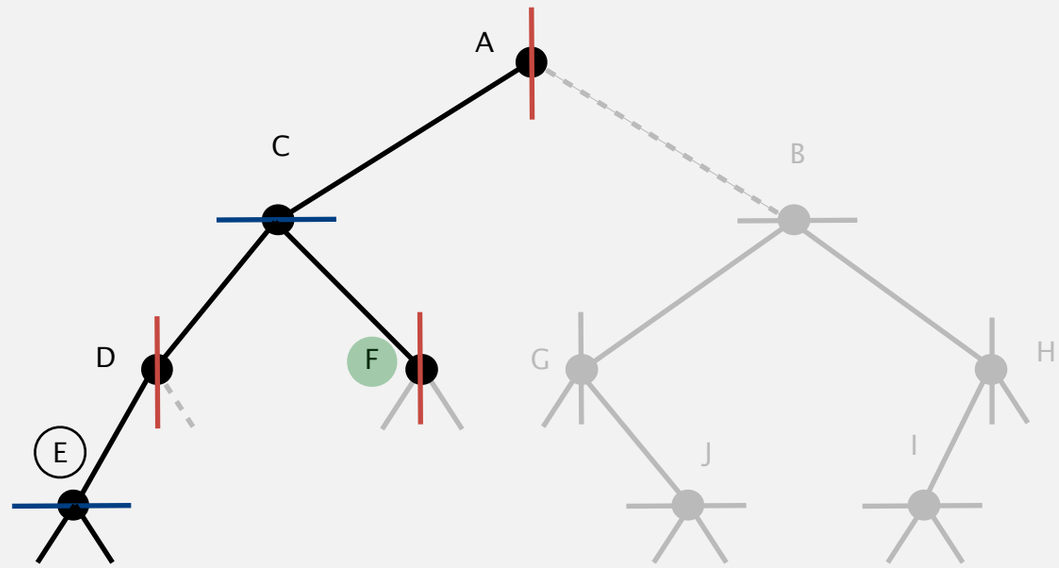
check if query rectangle contains point F

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





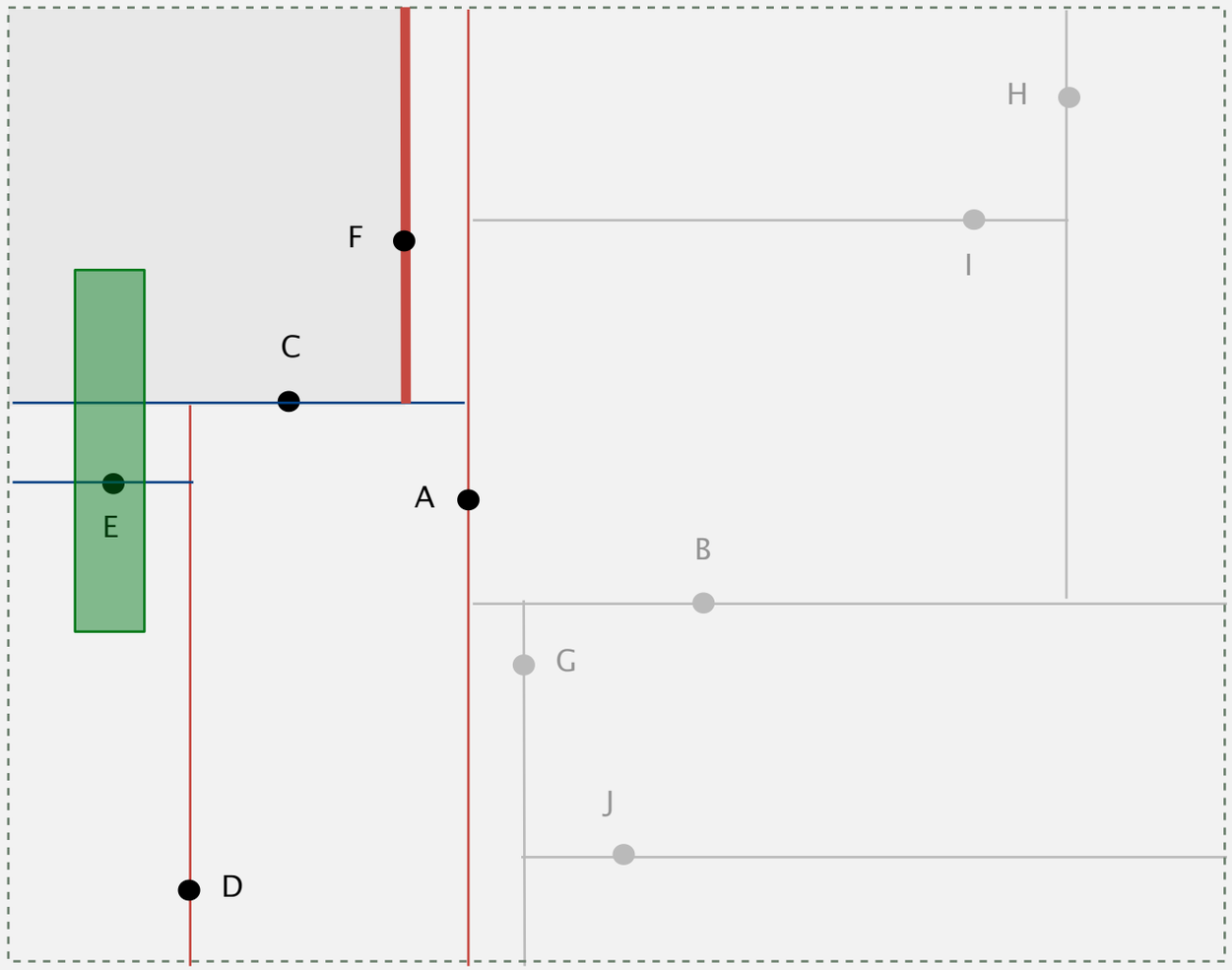
query rectangle to left of splitting line

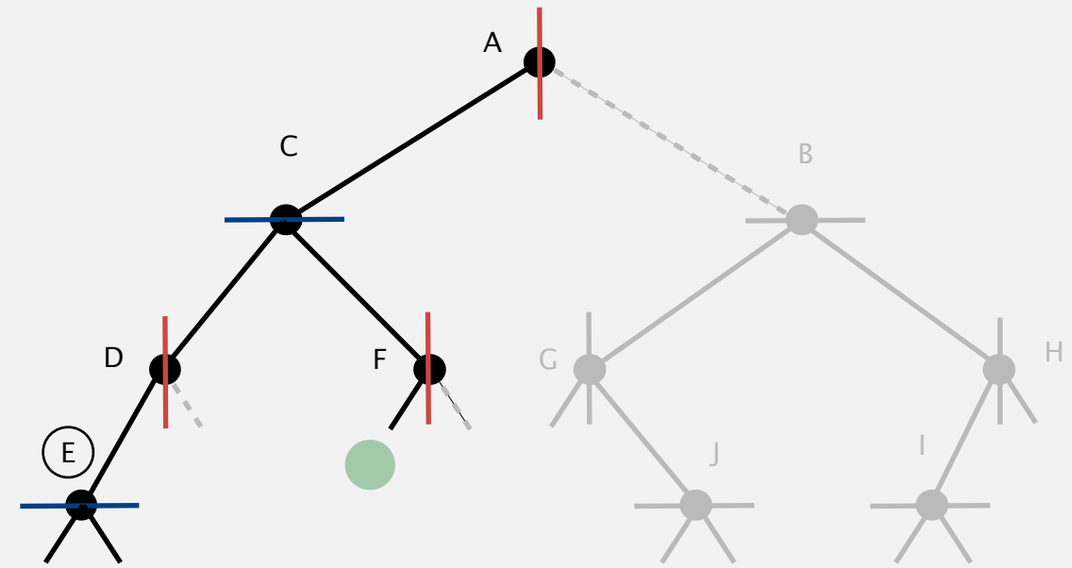
search only in left subtree

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





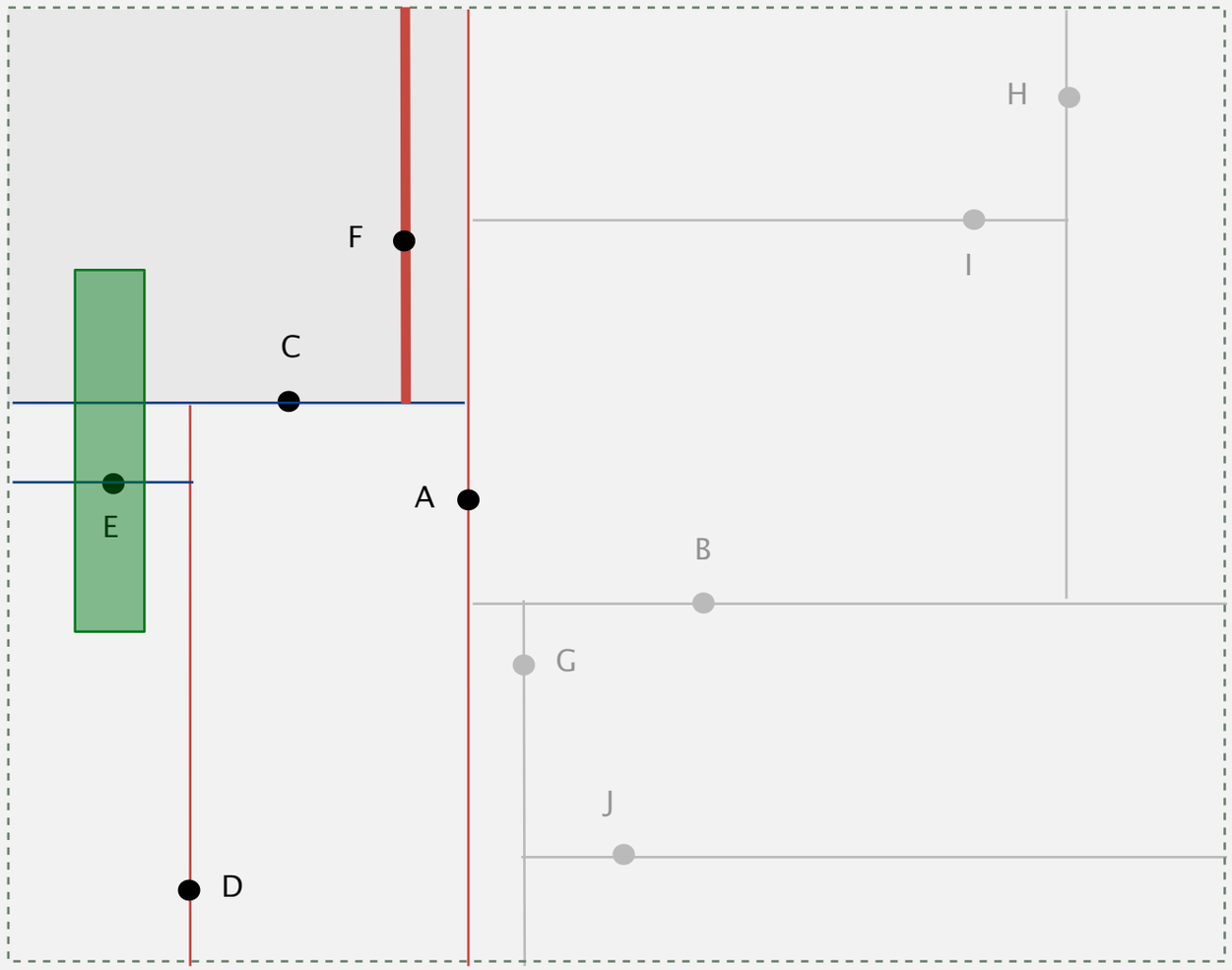
search left subtree

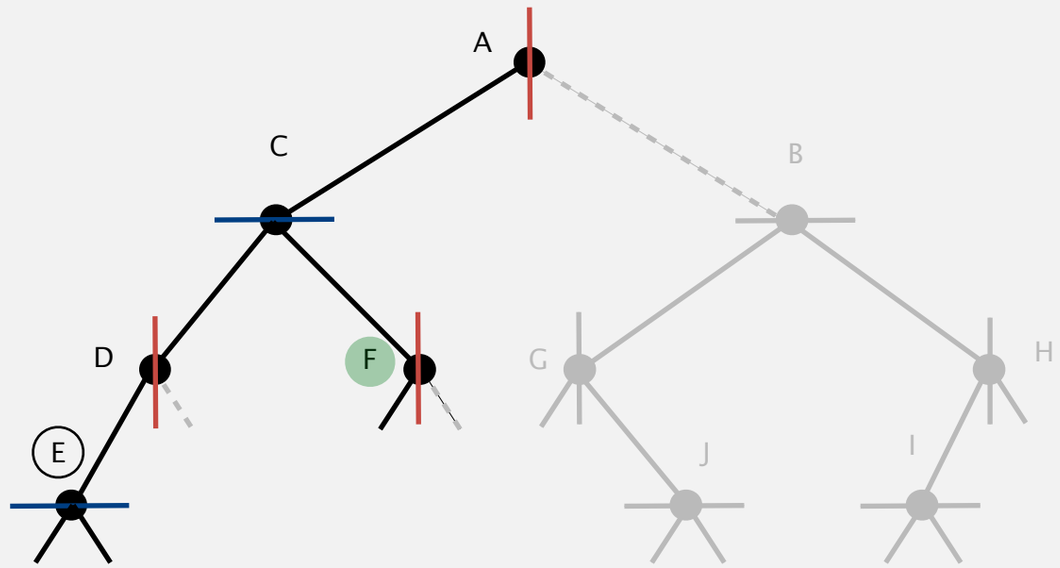
stop since empty

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.



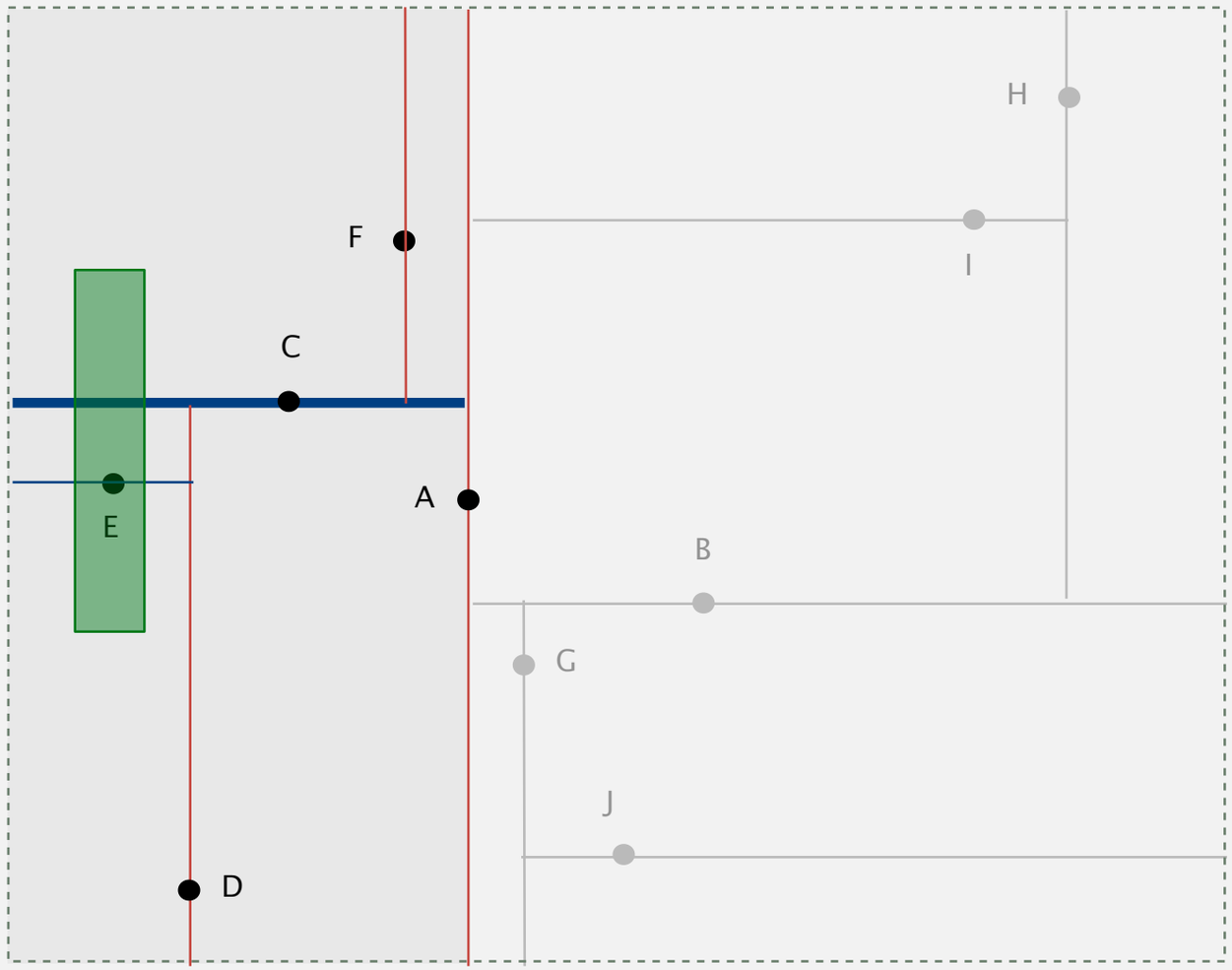


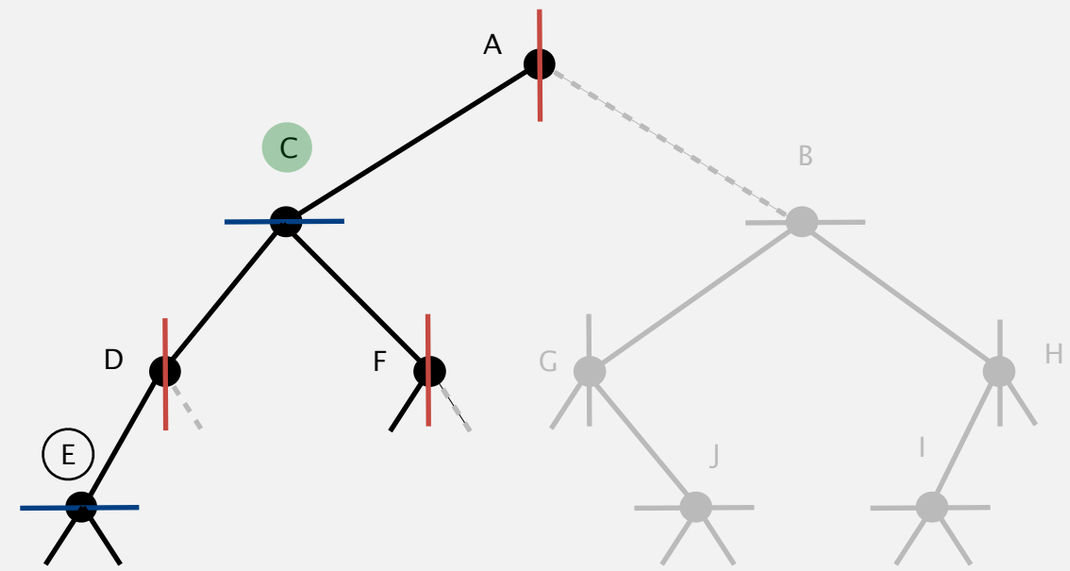
return from function call

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.



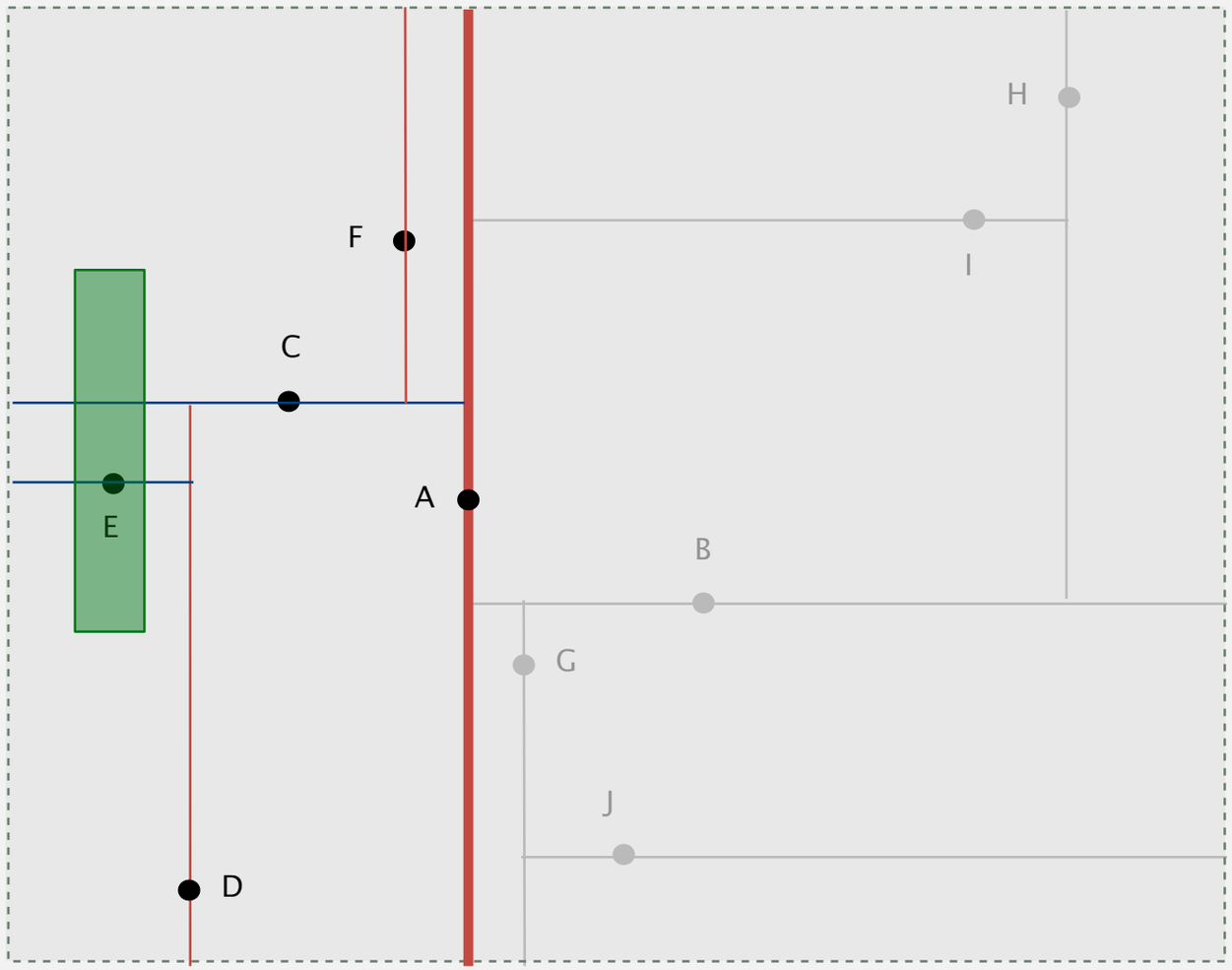


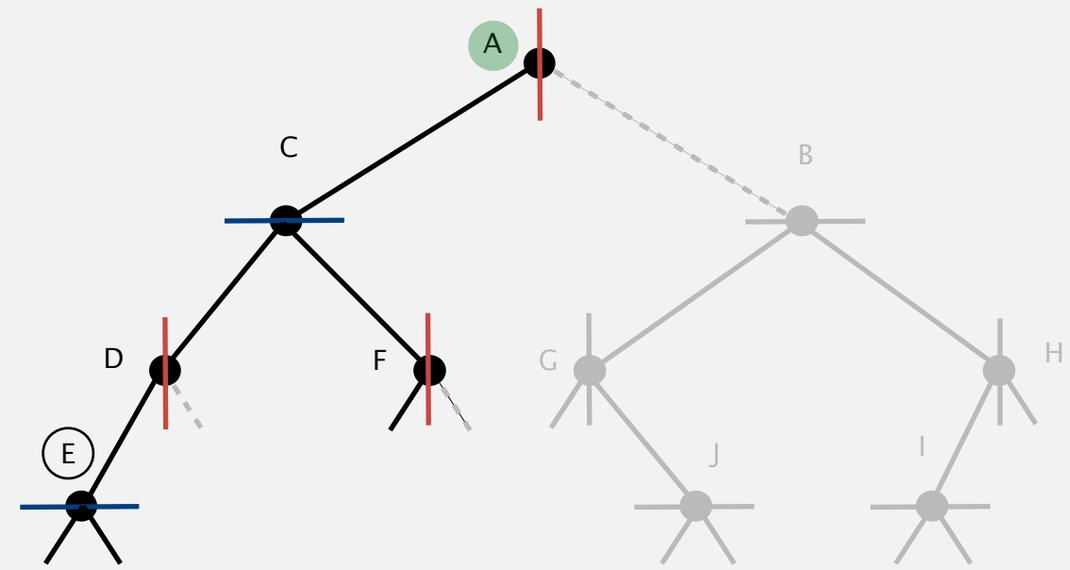
return from function call

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.



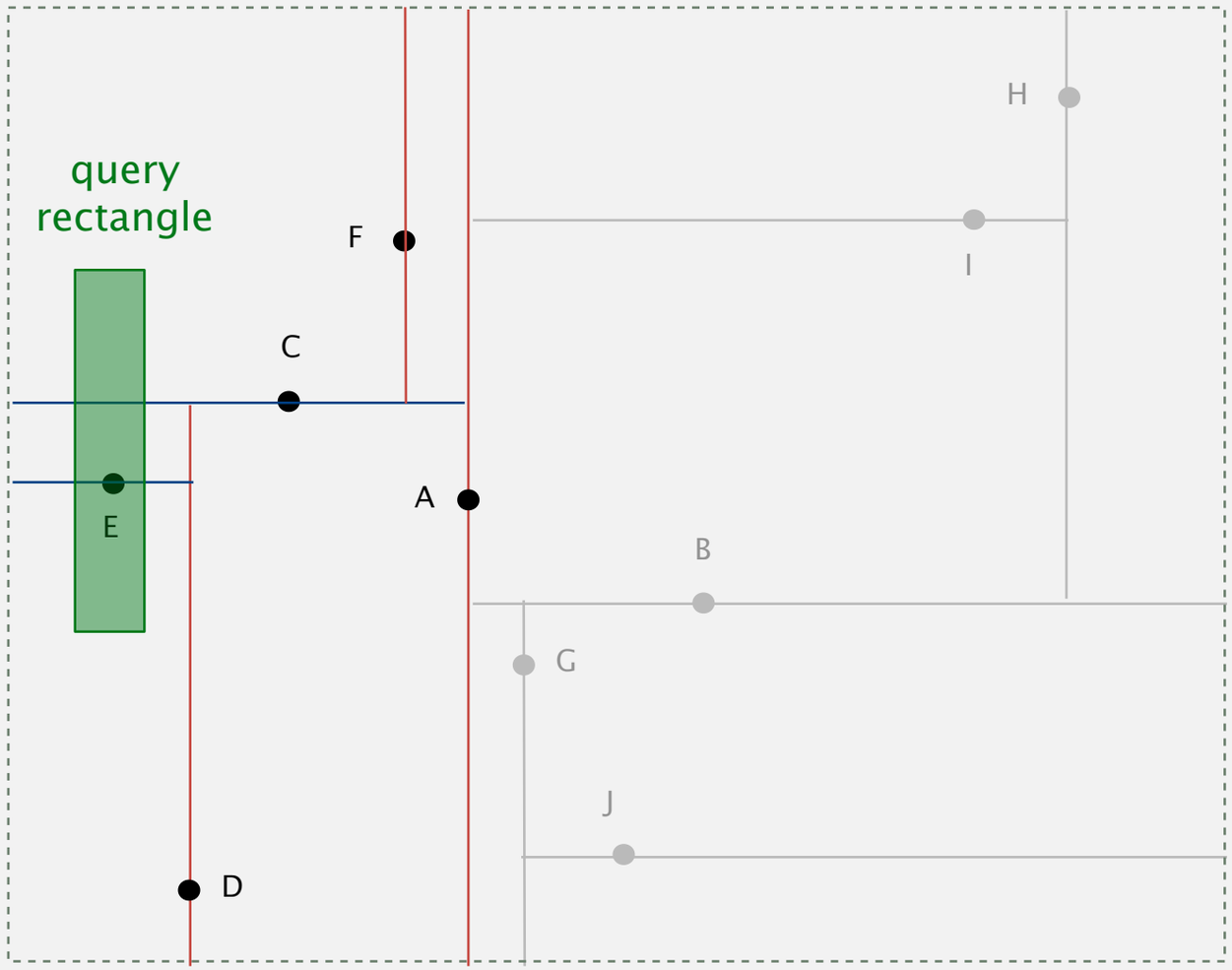


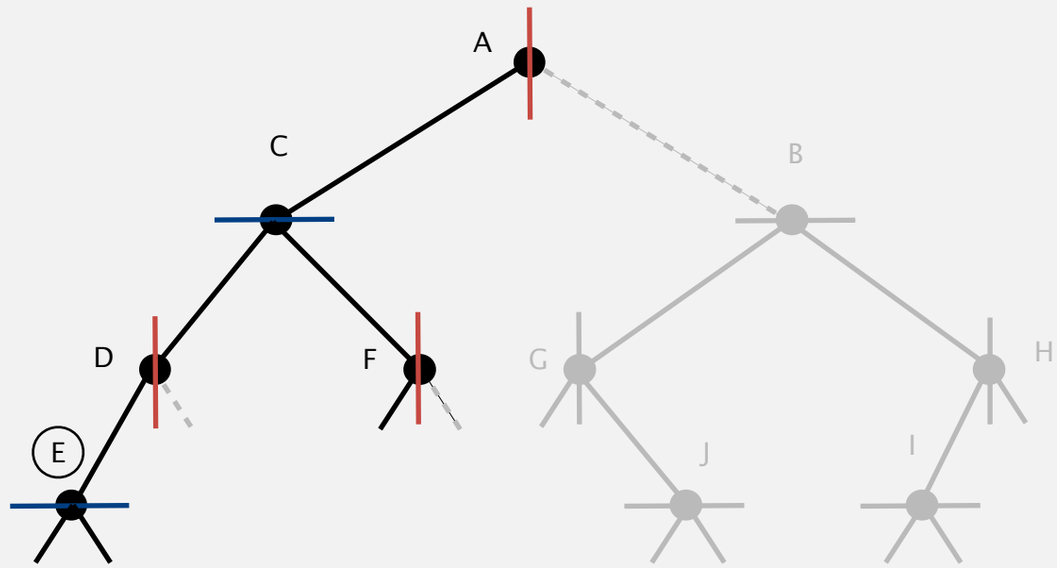
return from function call

2d tree demo: range search

Goal. Find all points in a query rectangle.

- Check if query rectangle contains point in node.
- Recursively search left/bottom and right/top subtrees.
- Optimization: prune subtree if it can't contain a point in rectangle.





done

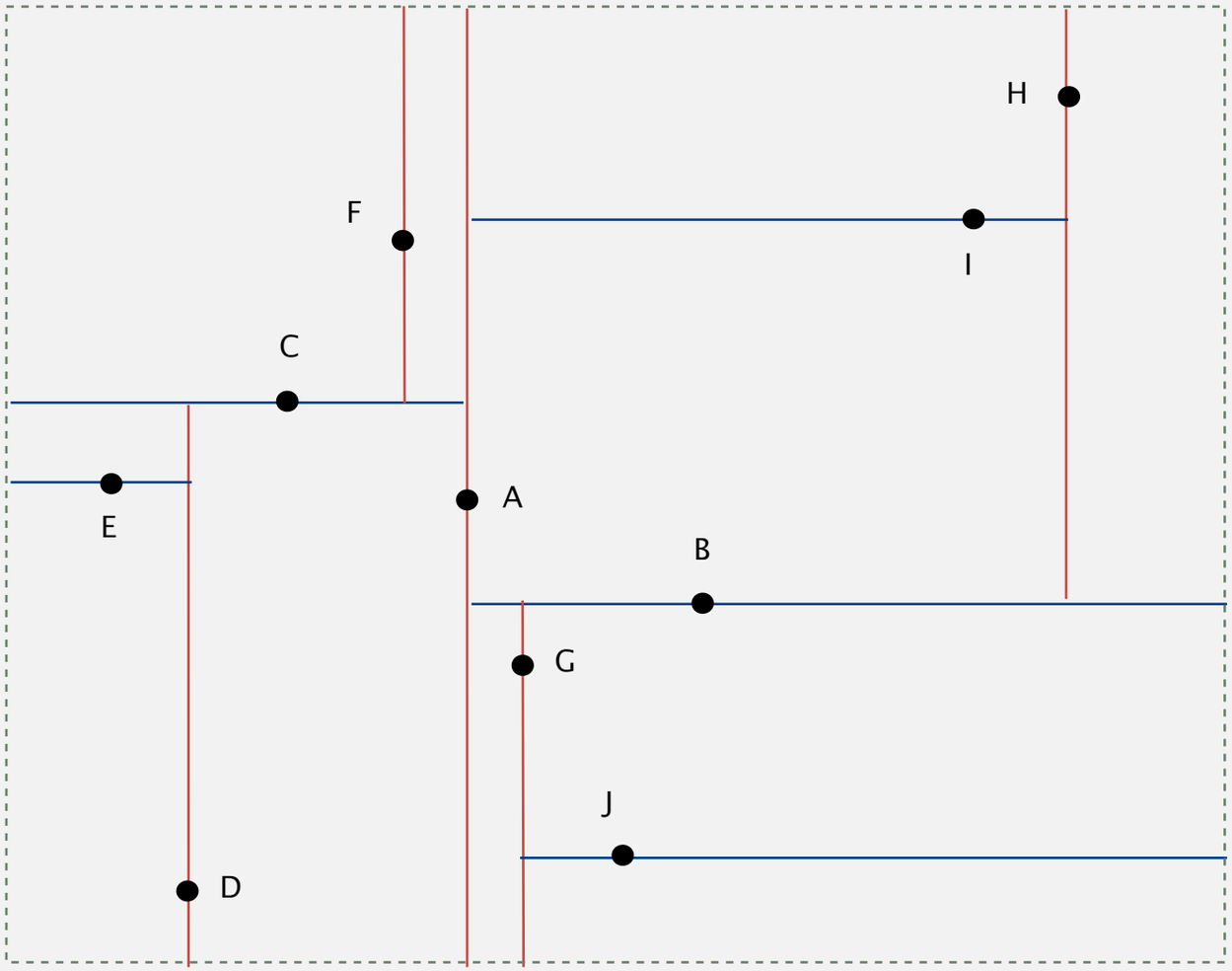
https://algs4.cs.princeton.edu

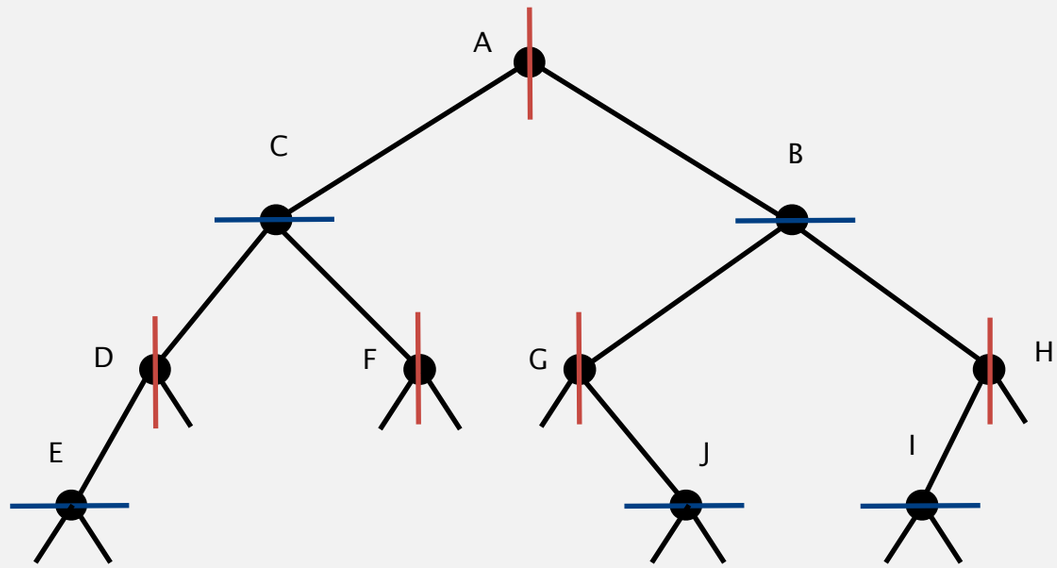
2d Tree Demo

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

2d tree demo: nearest neighbor

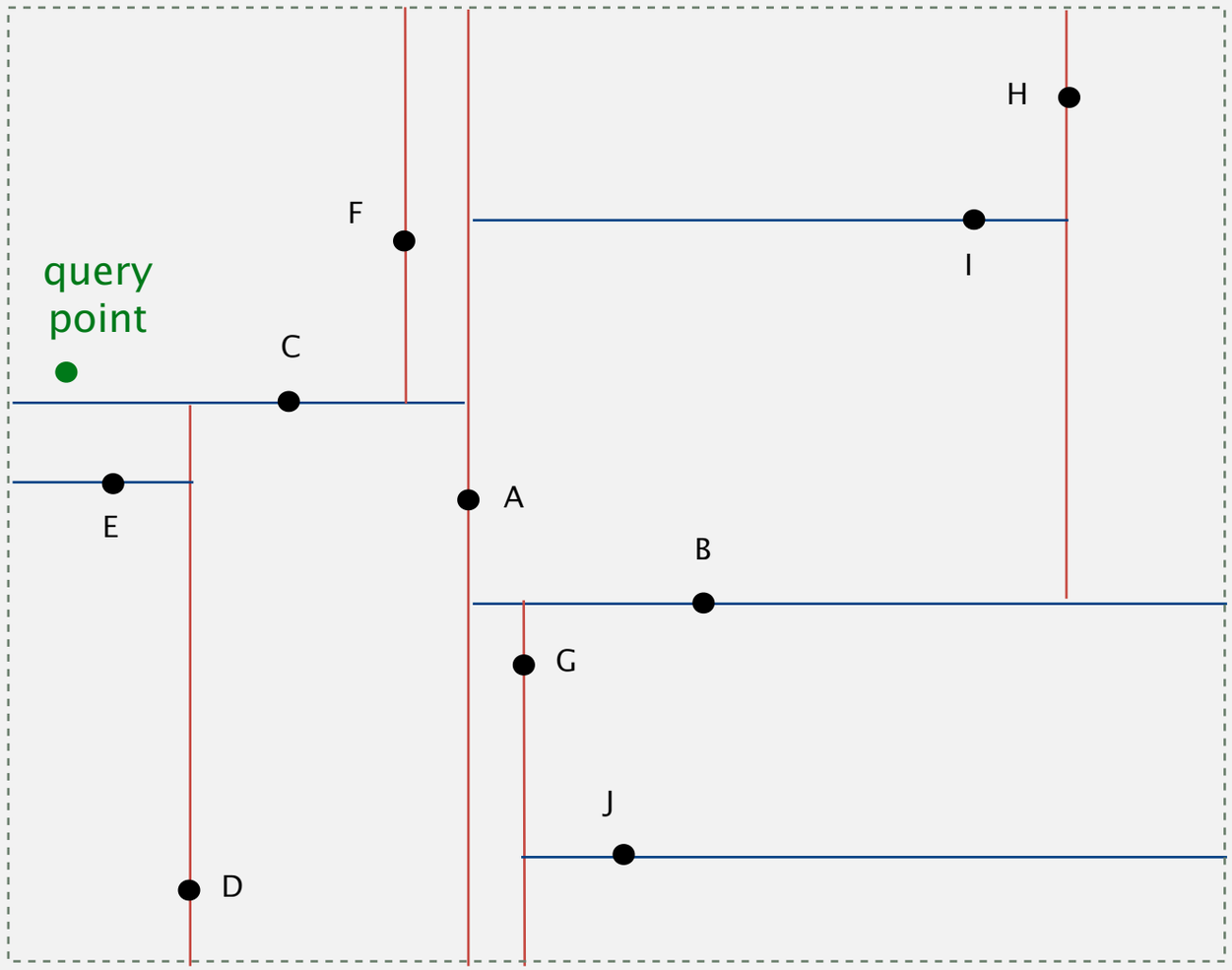
Goal. Find closest point to query point.

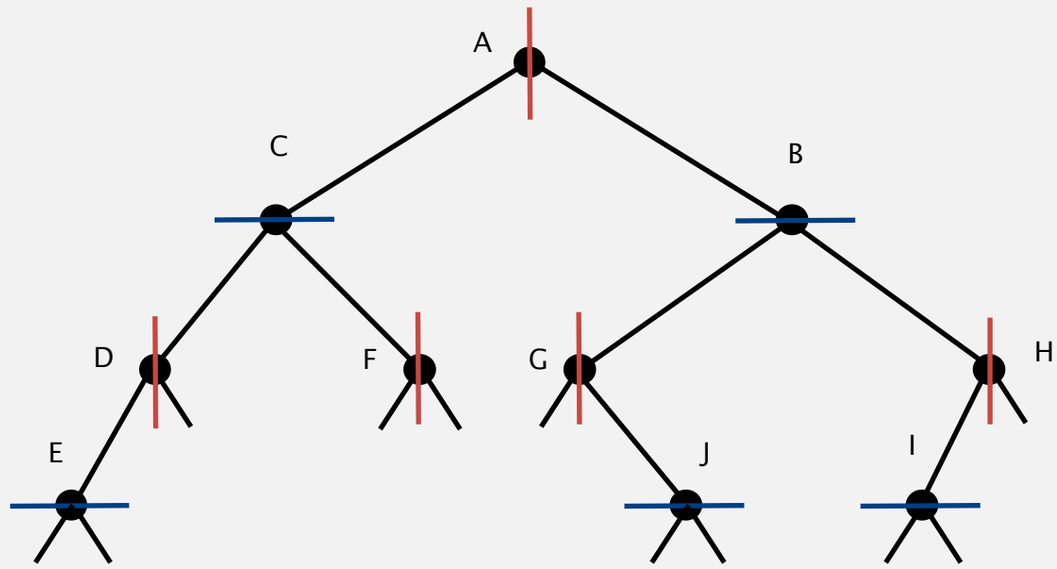




2d tree demo: nearest neighbor

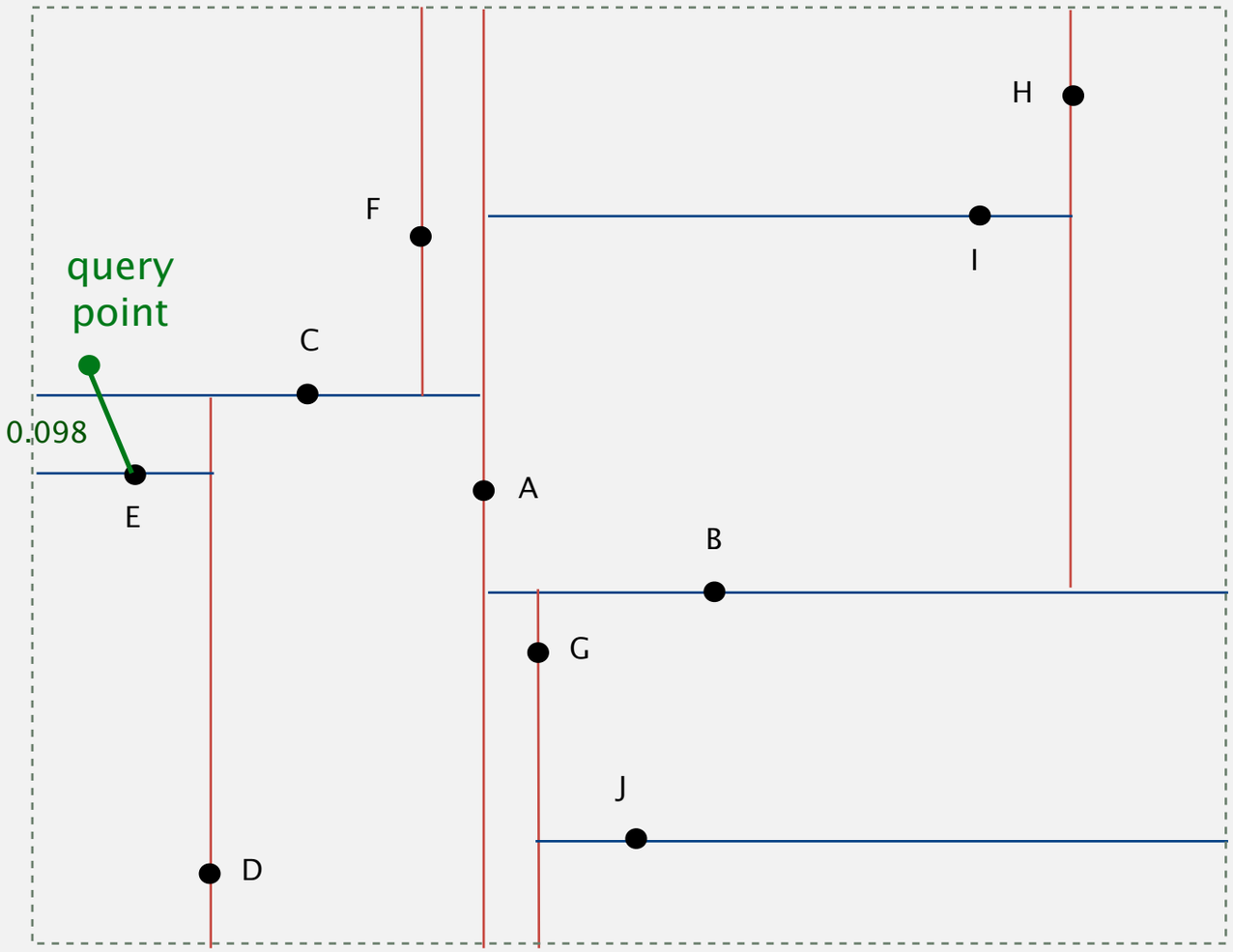
Goal. Find closest point to query point.

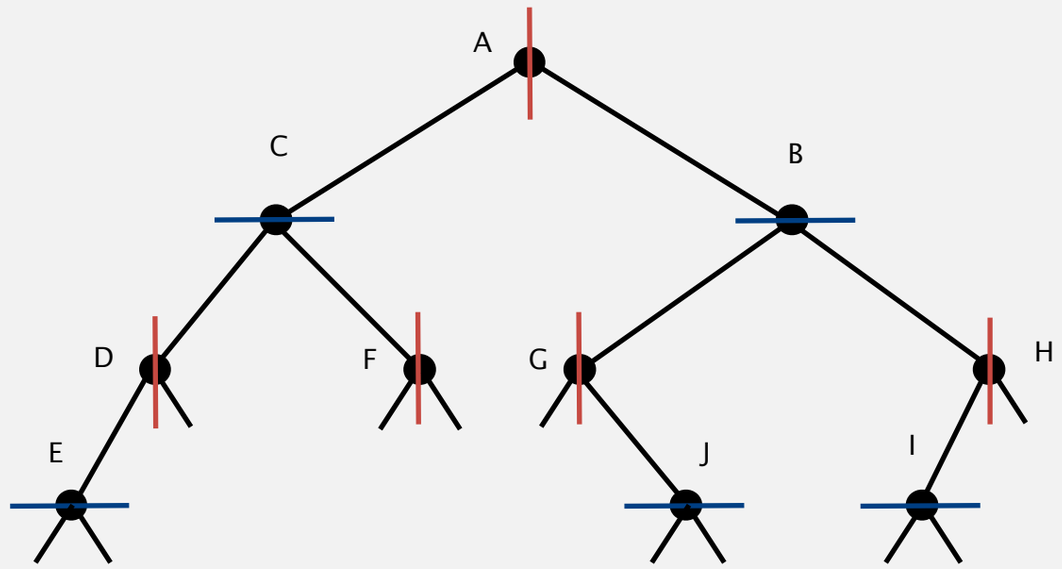




2d tree demo: nearest neighbor

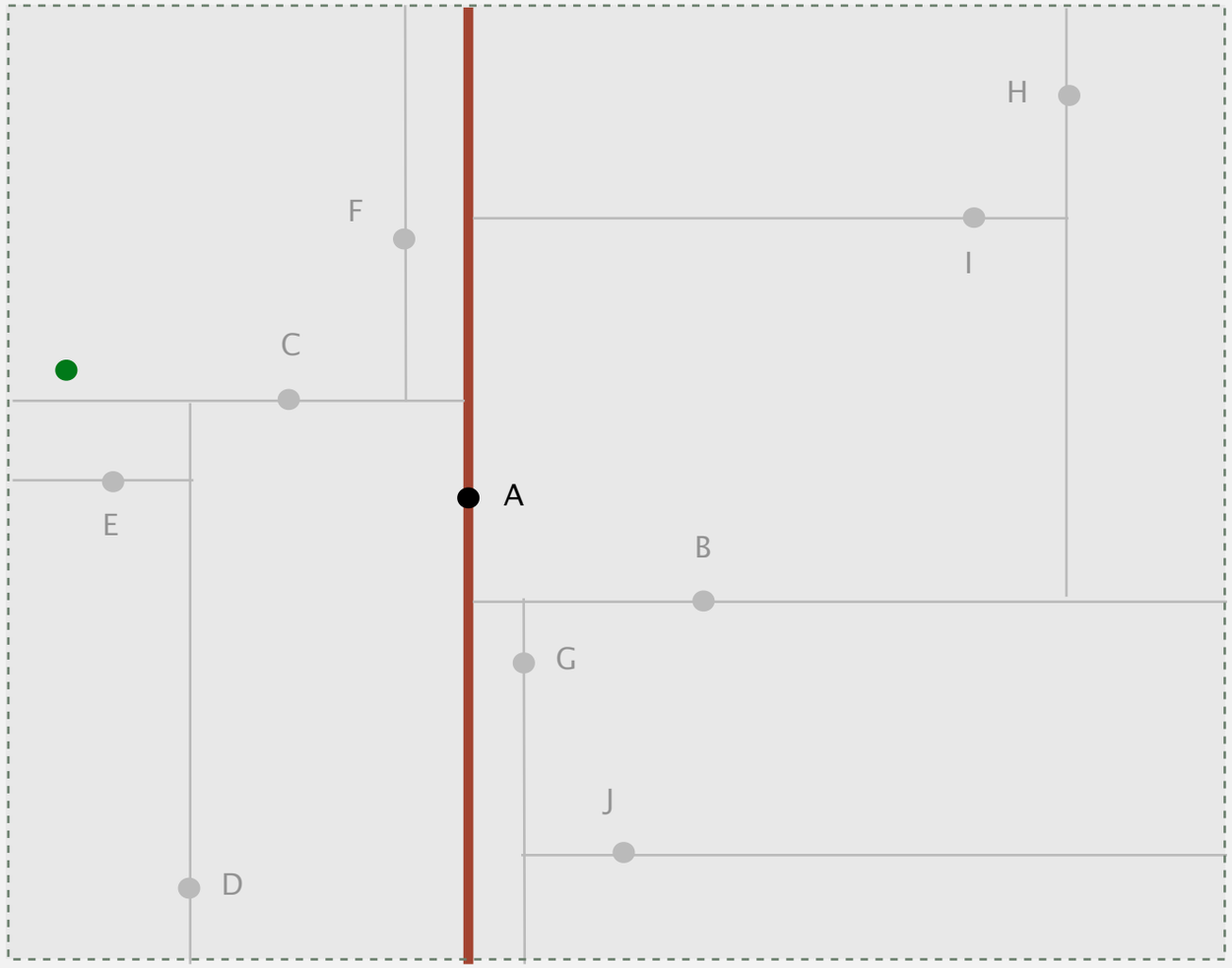
Goal. Find closest point to query point.

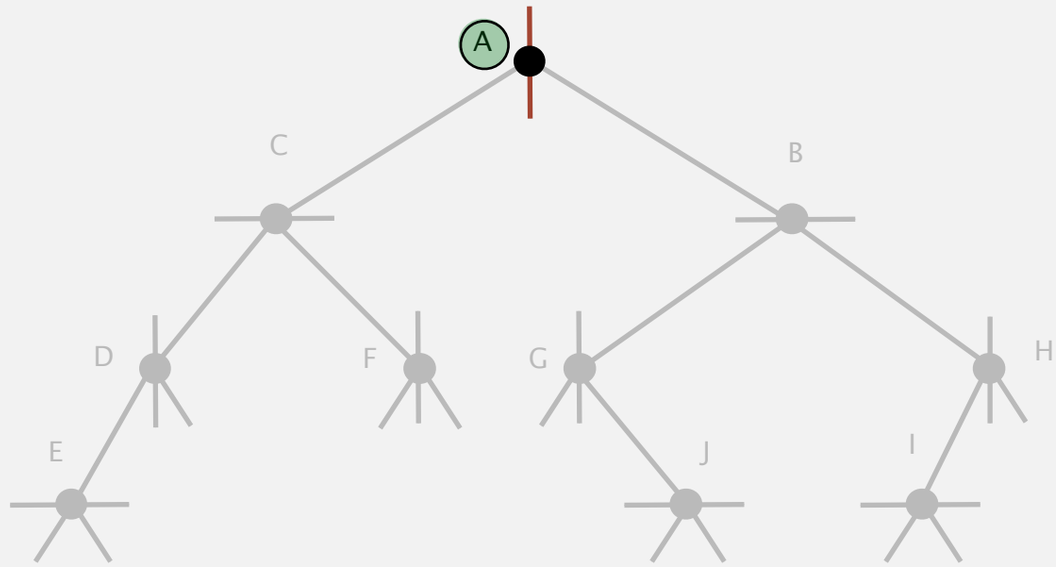




2d tree demo: nearest neighbor

- Check distance from point in node to query point.





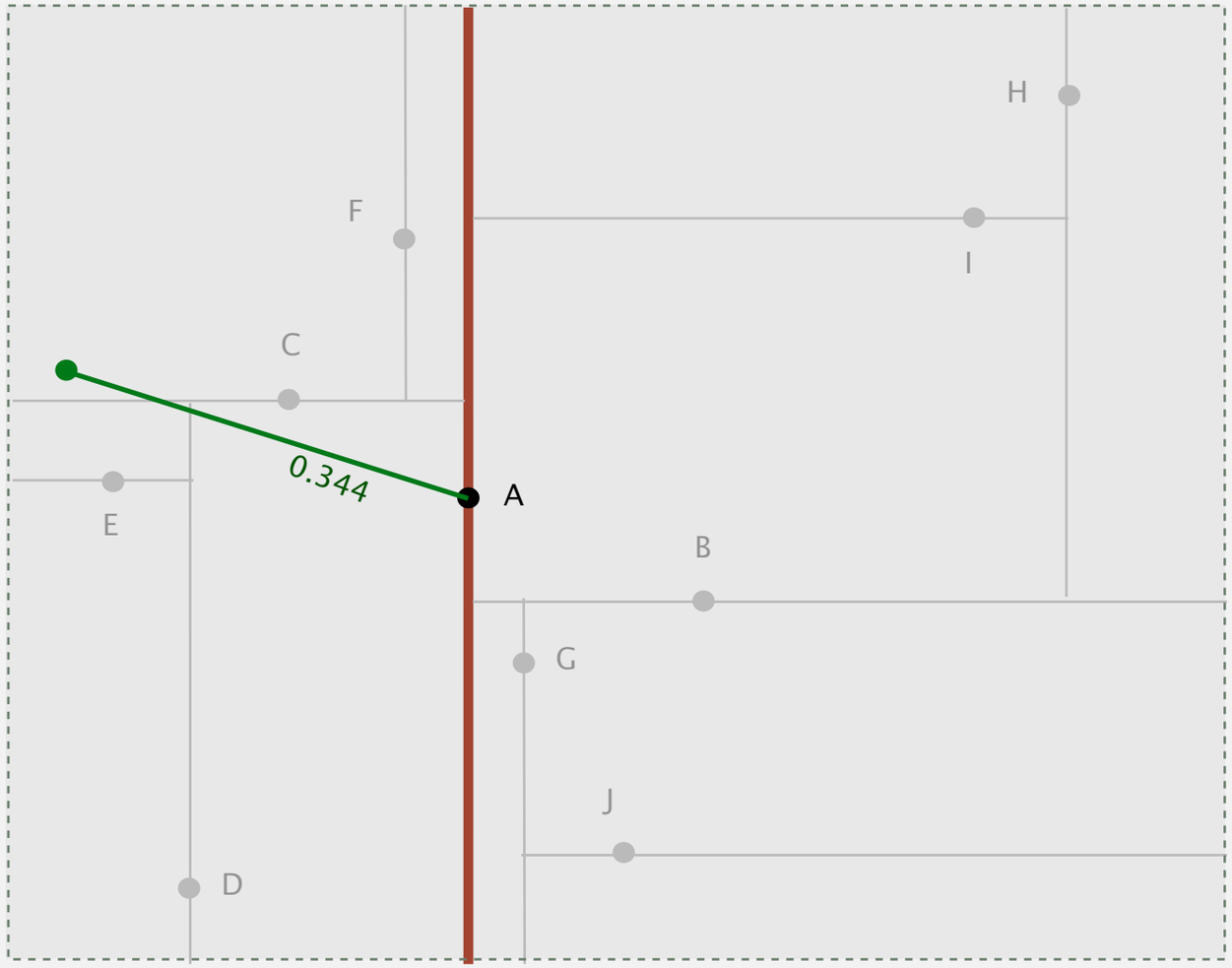
search root node

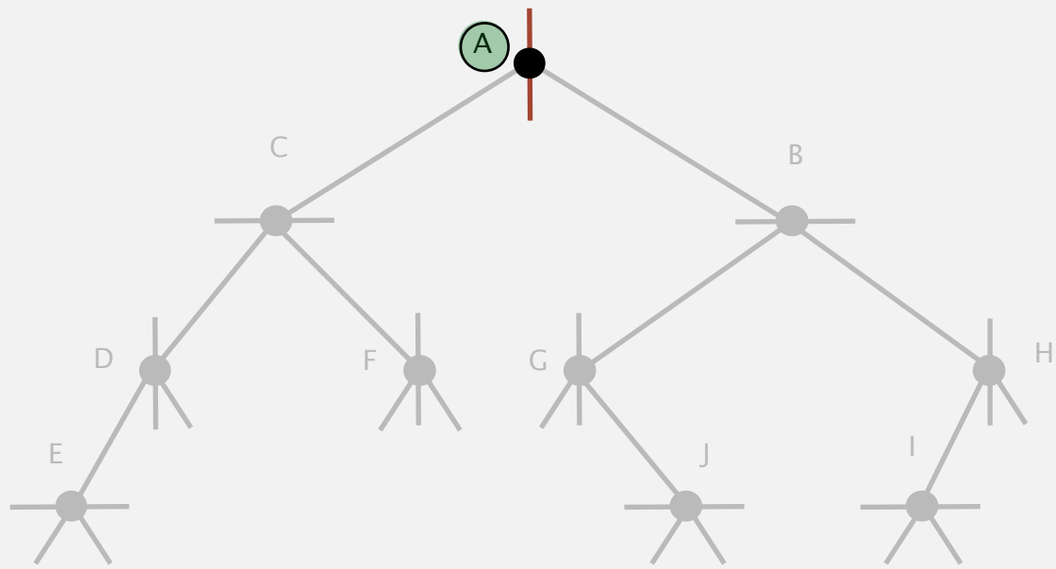
compute distance from query point to A

(update champion nearest neighbor)

2d tree demo: nearest neighbor

- Check distance from point in node to query point.





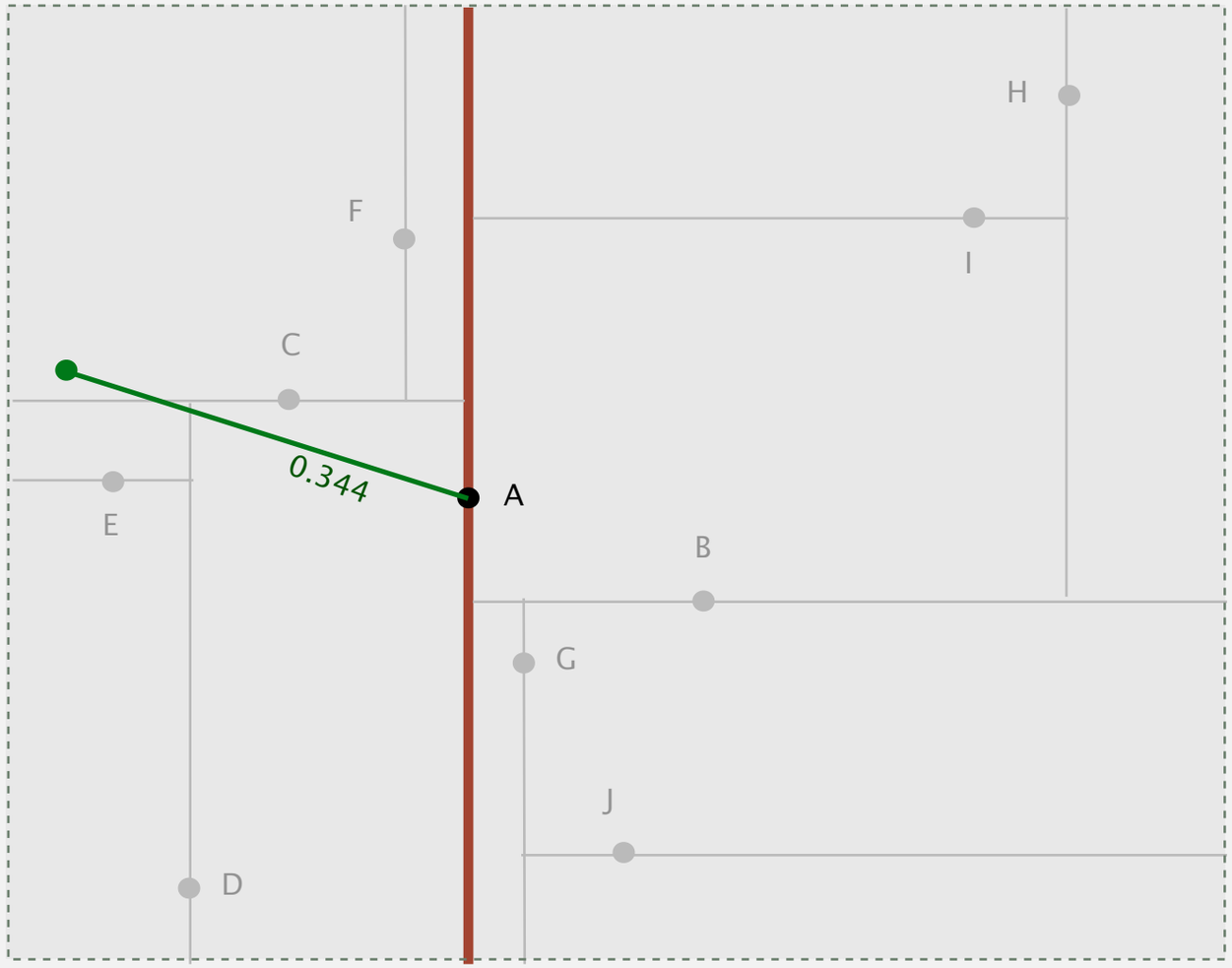
search root node

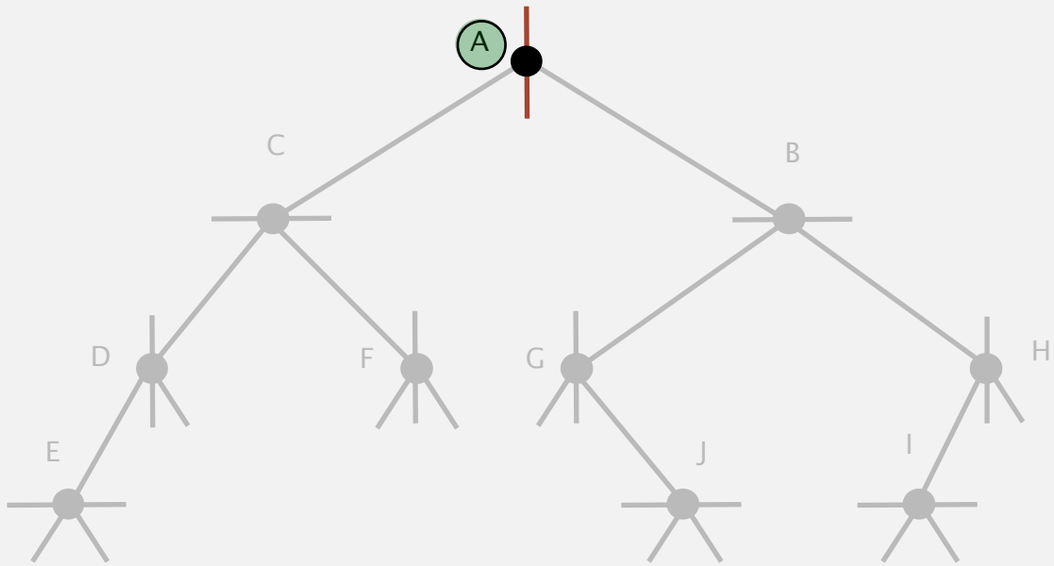
compute distance from query point to A

(update champion nearest neighbor)

2d tree demo: nearest neighbor

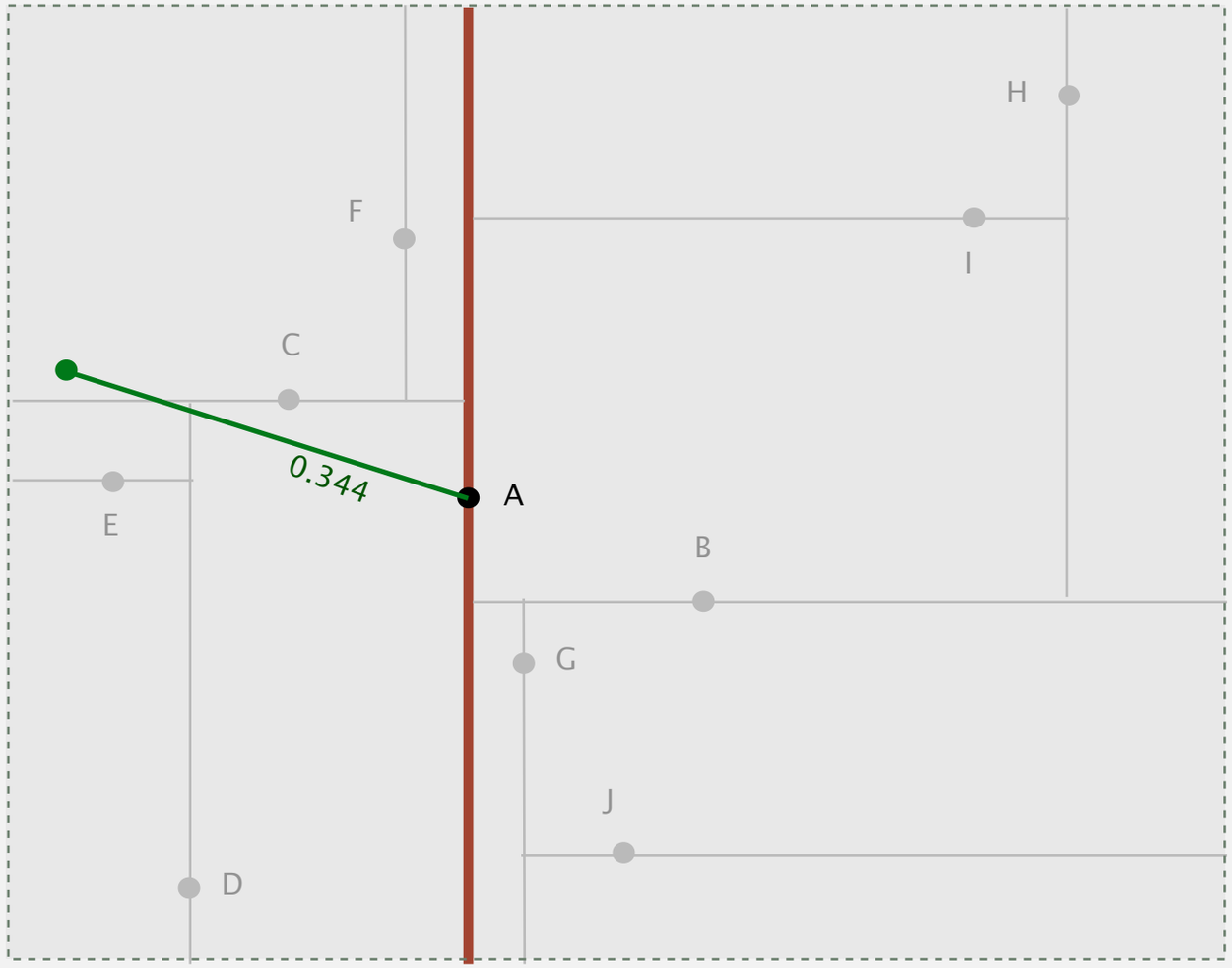
- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.

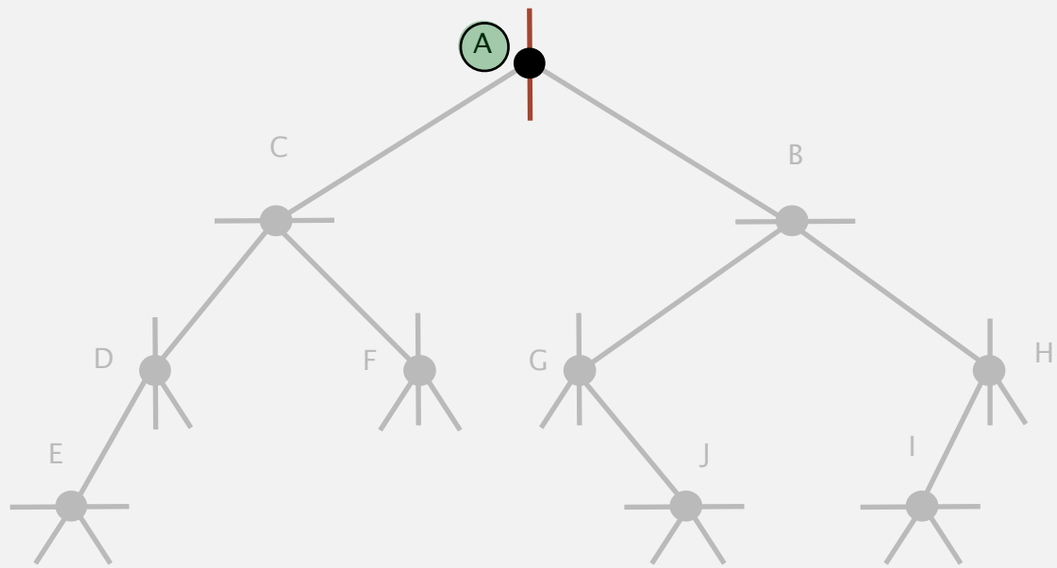




2d tree demo: nearest neighbor

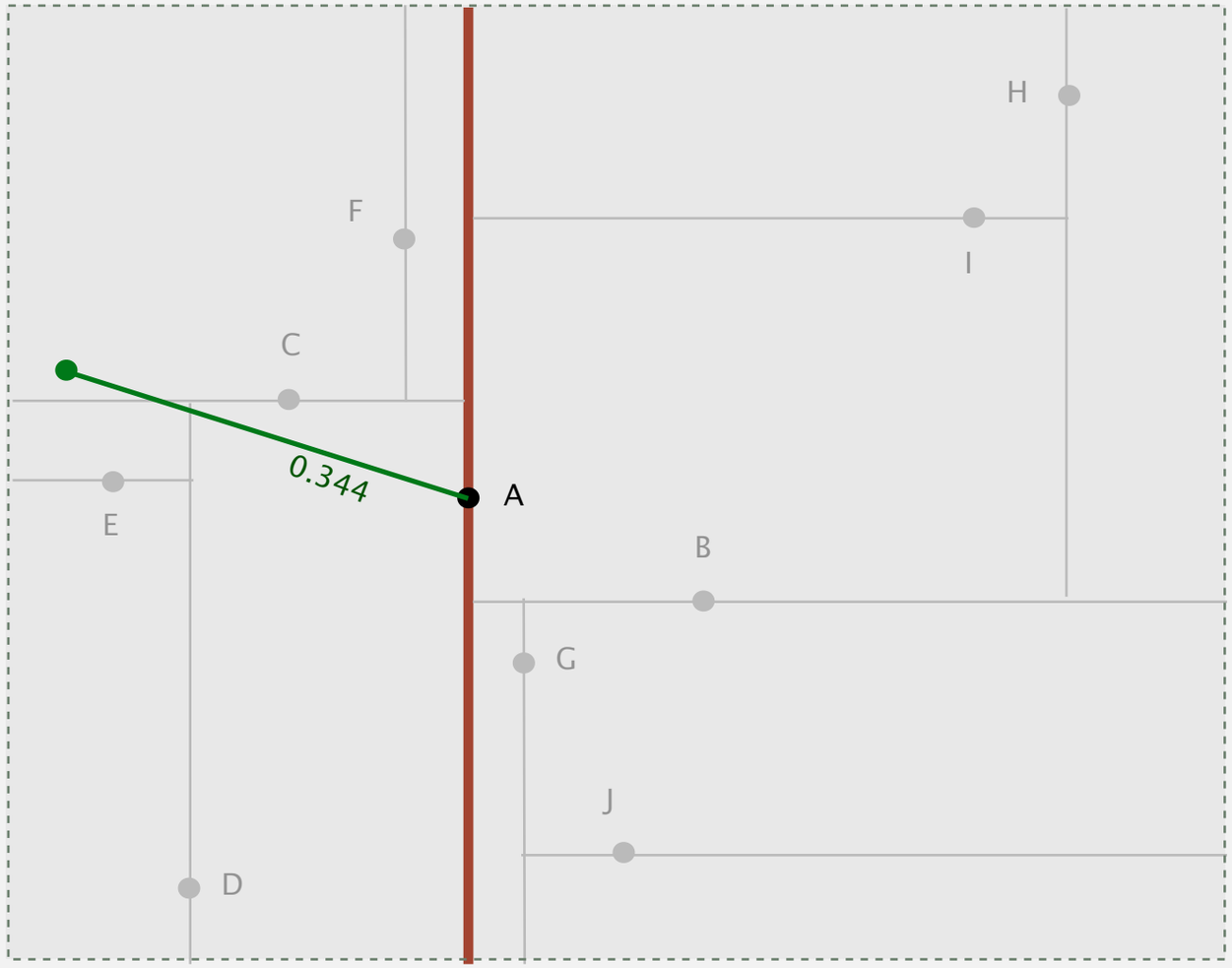
- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.

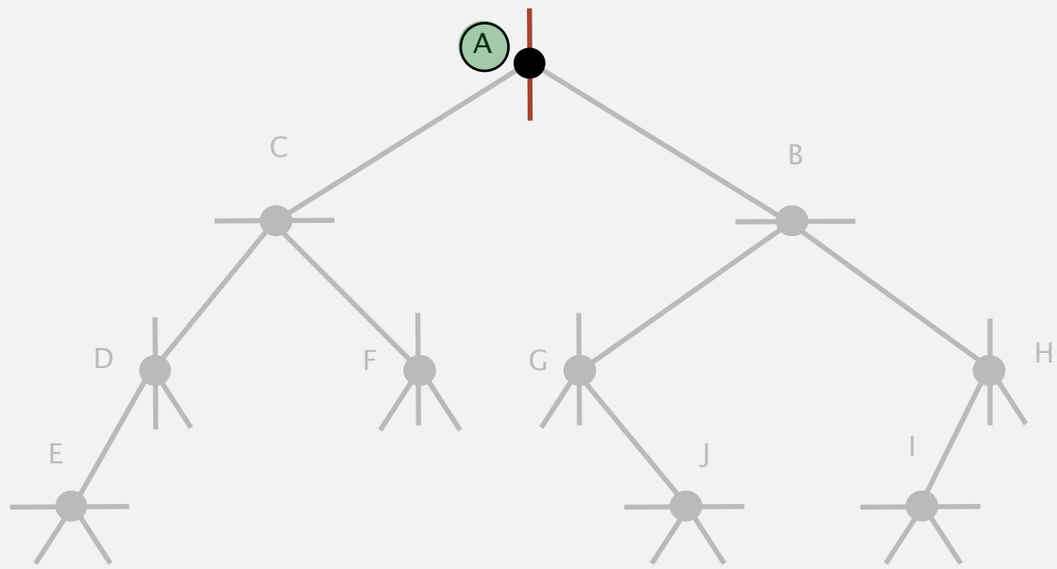




2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



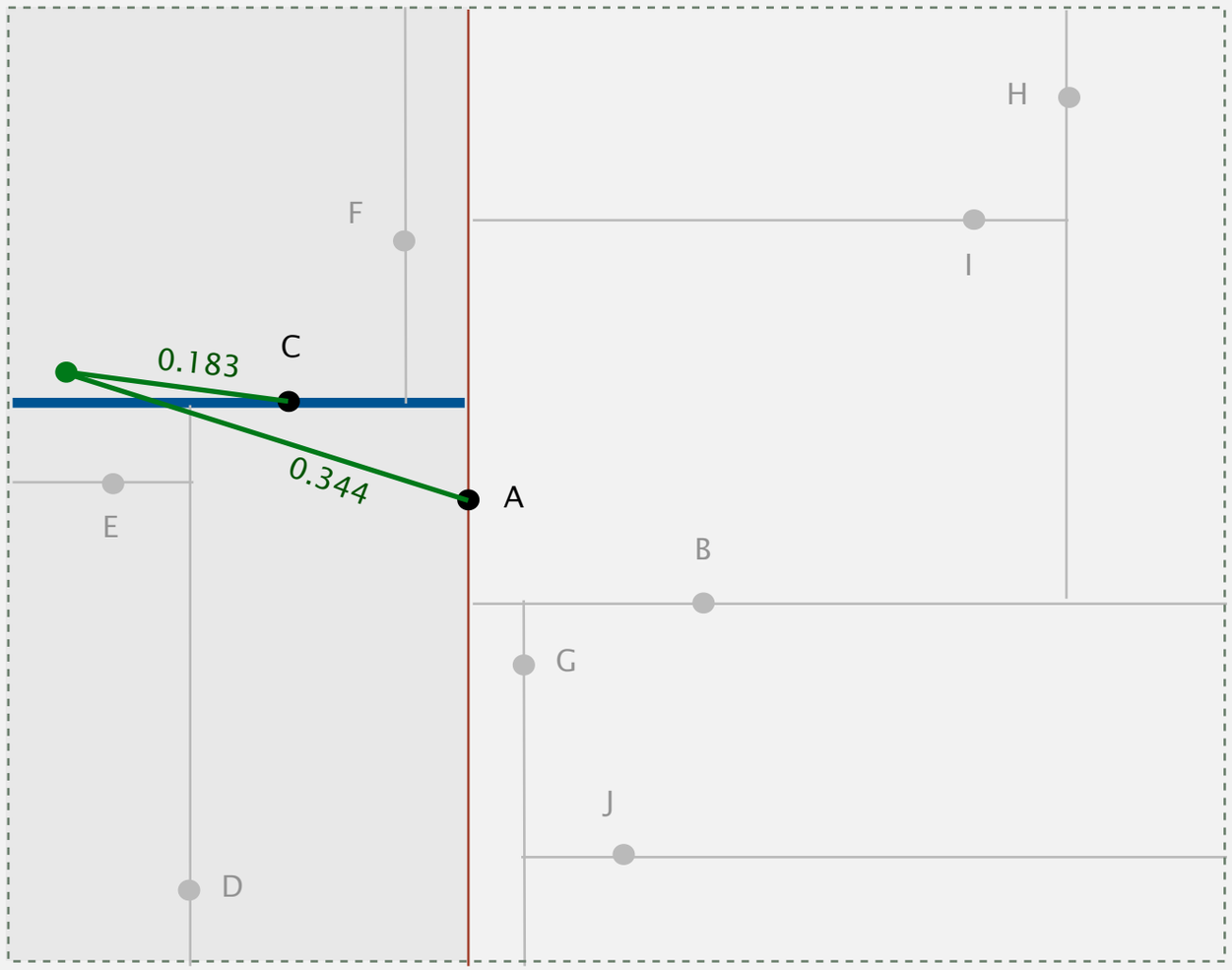


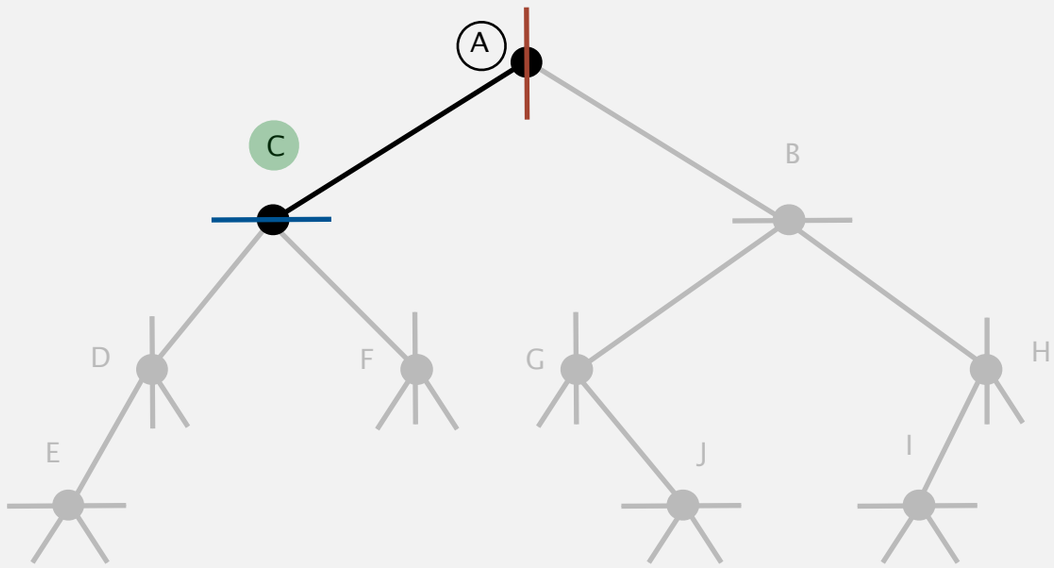
query point is to the left of splitting line

search left subtree first

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.





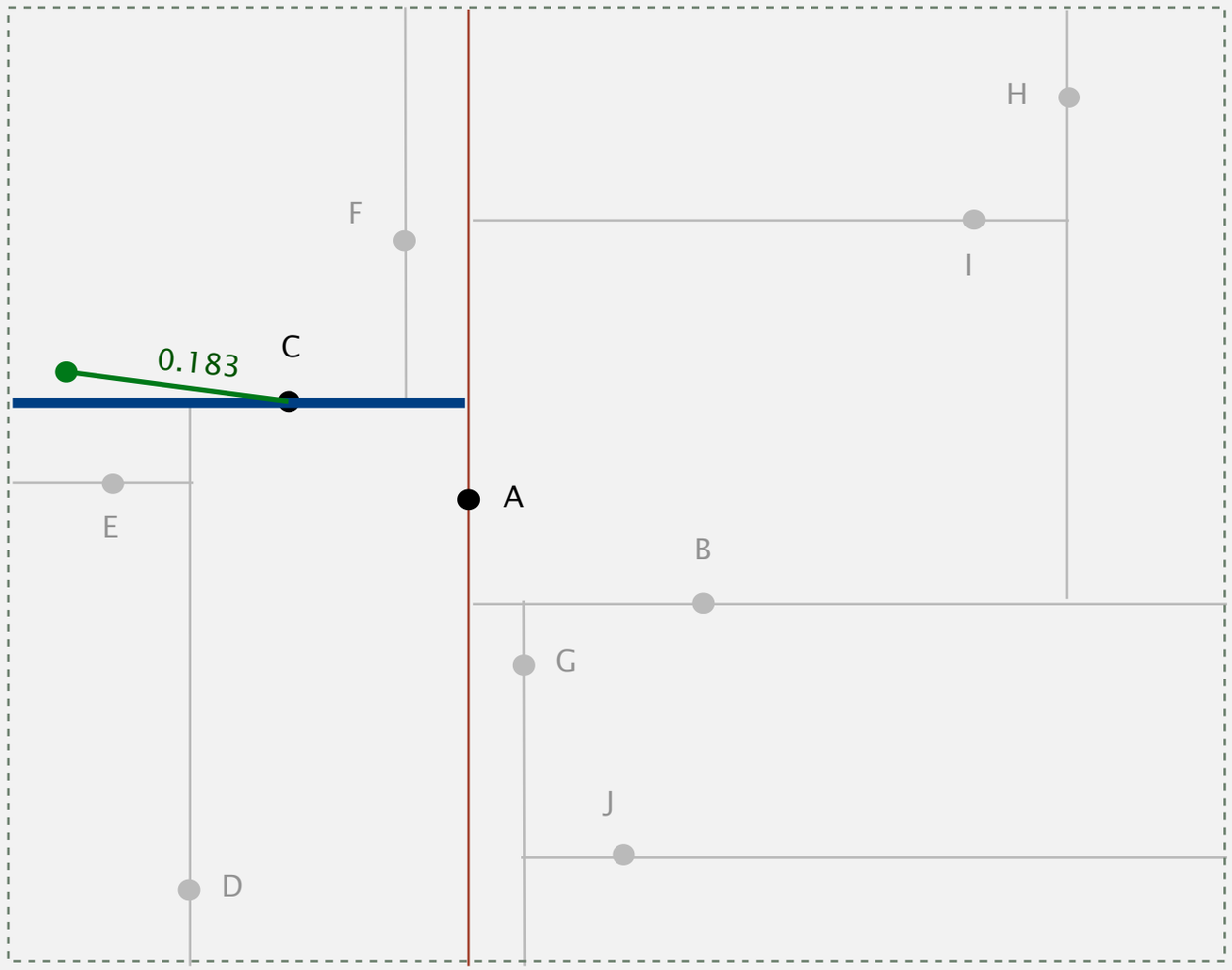
search left subtree

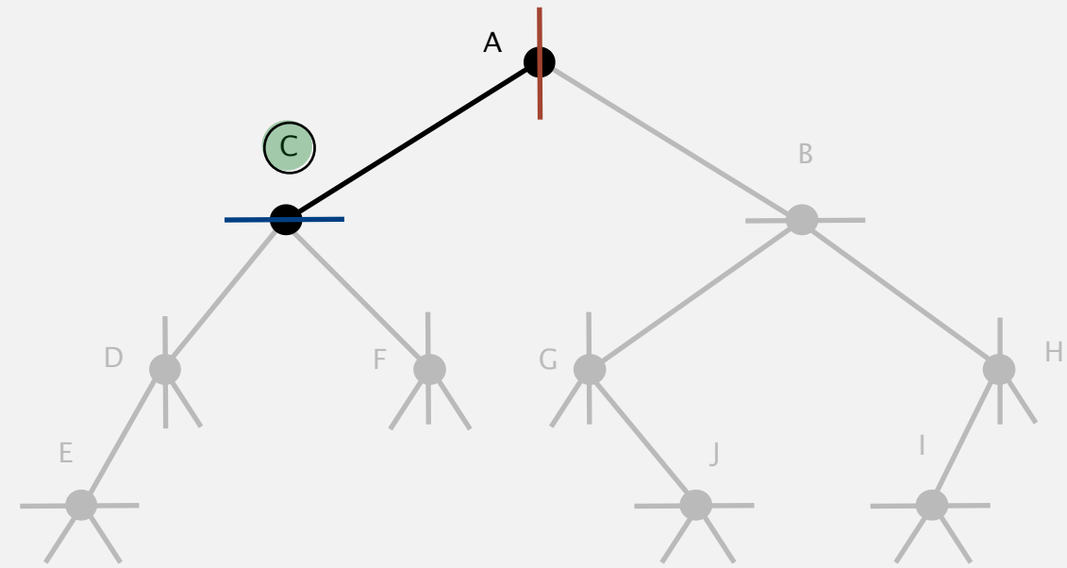
compute distance from query point to C

(update champion)

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



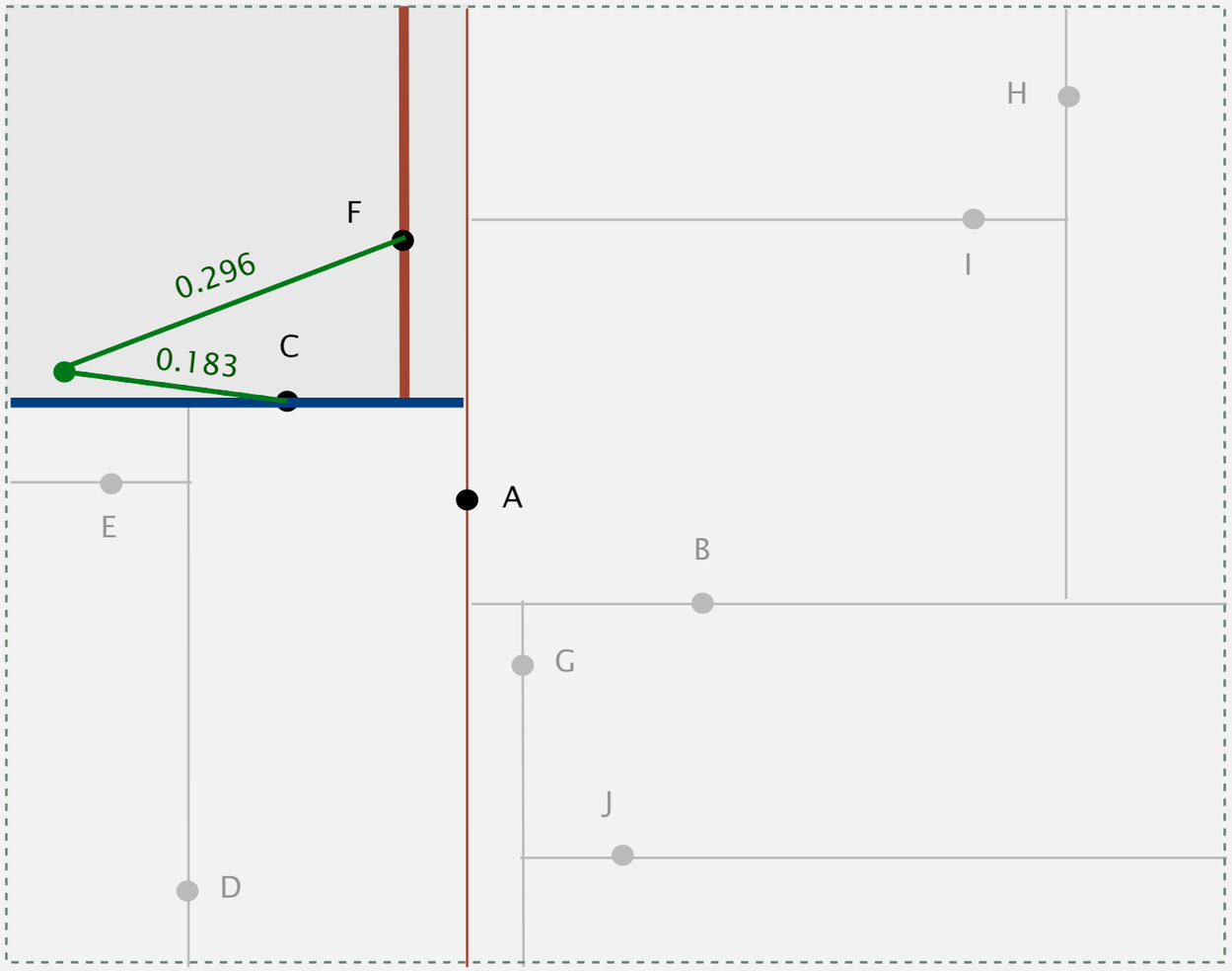


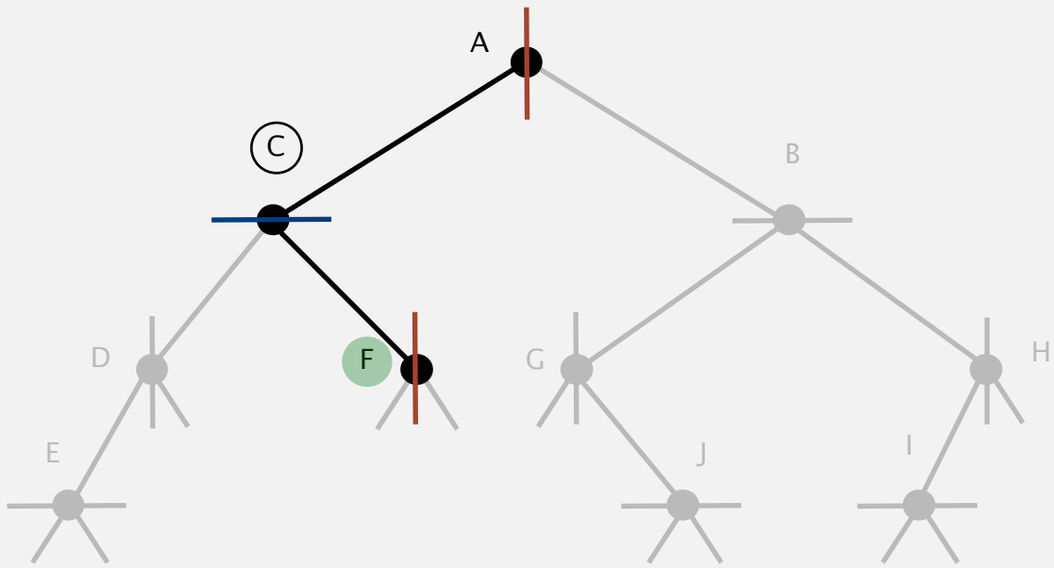
query point is above splitting line

search top subtree first

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



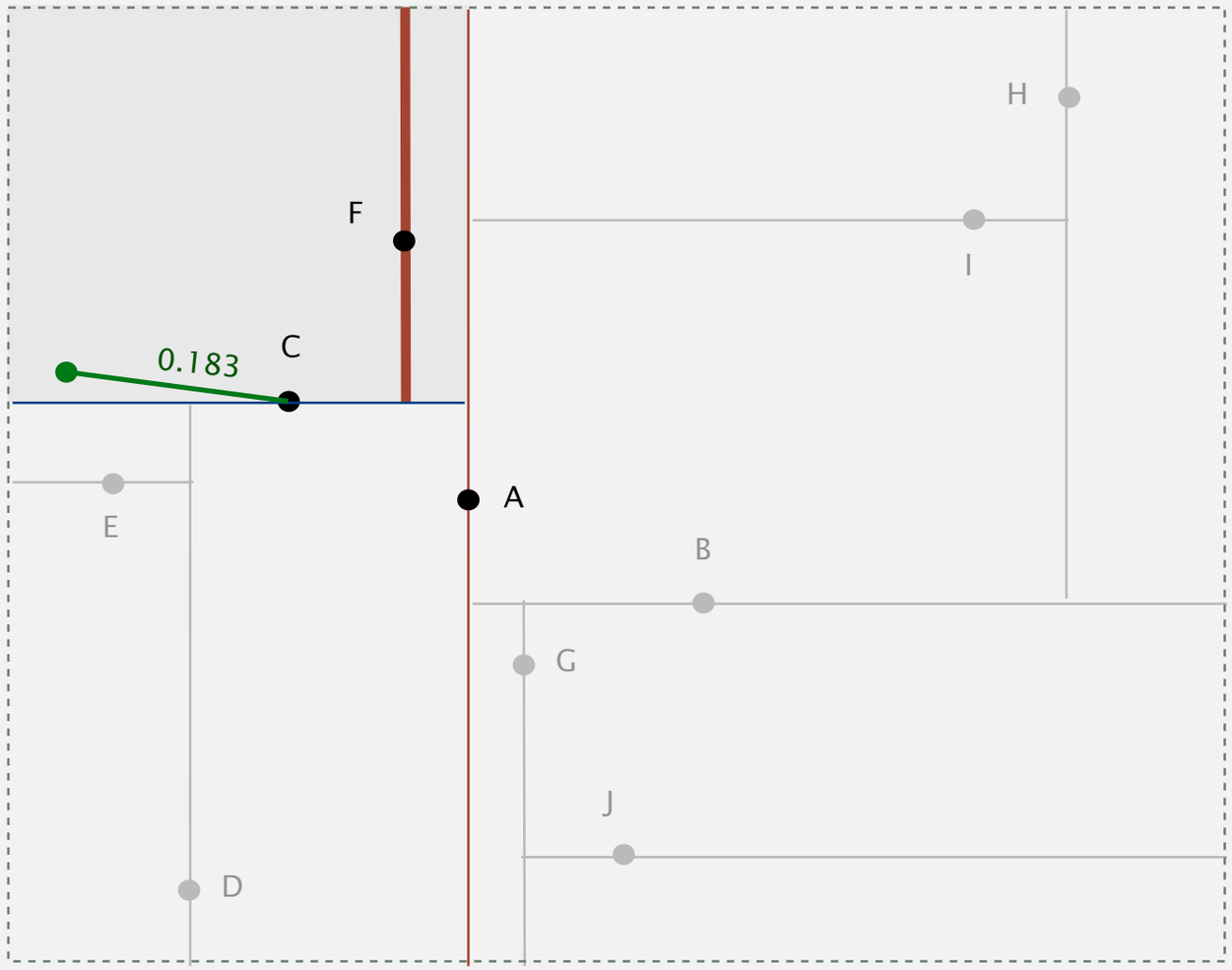


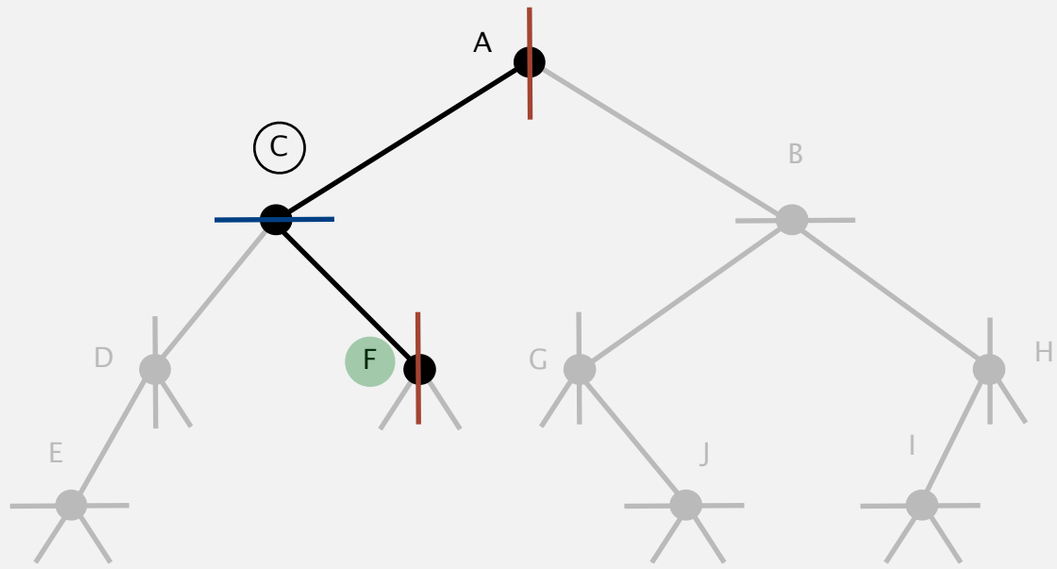
search top subtree

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 and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



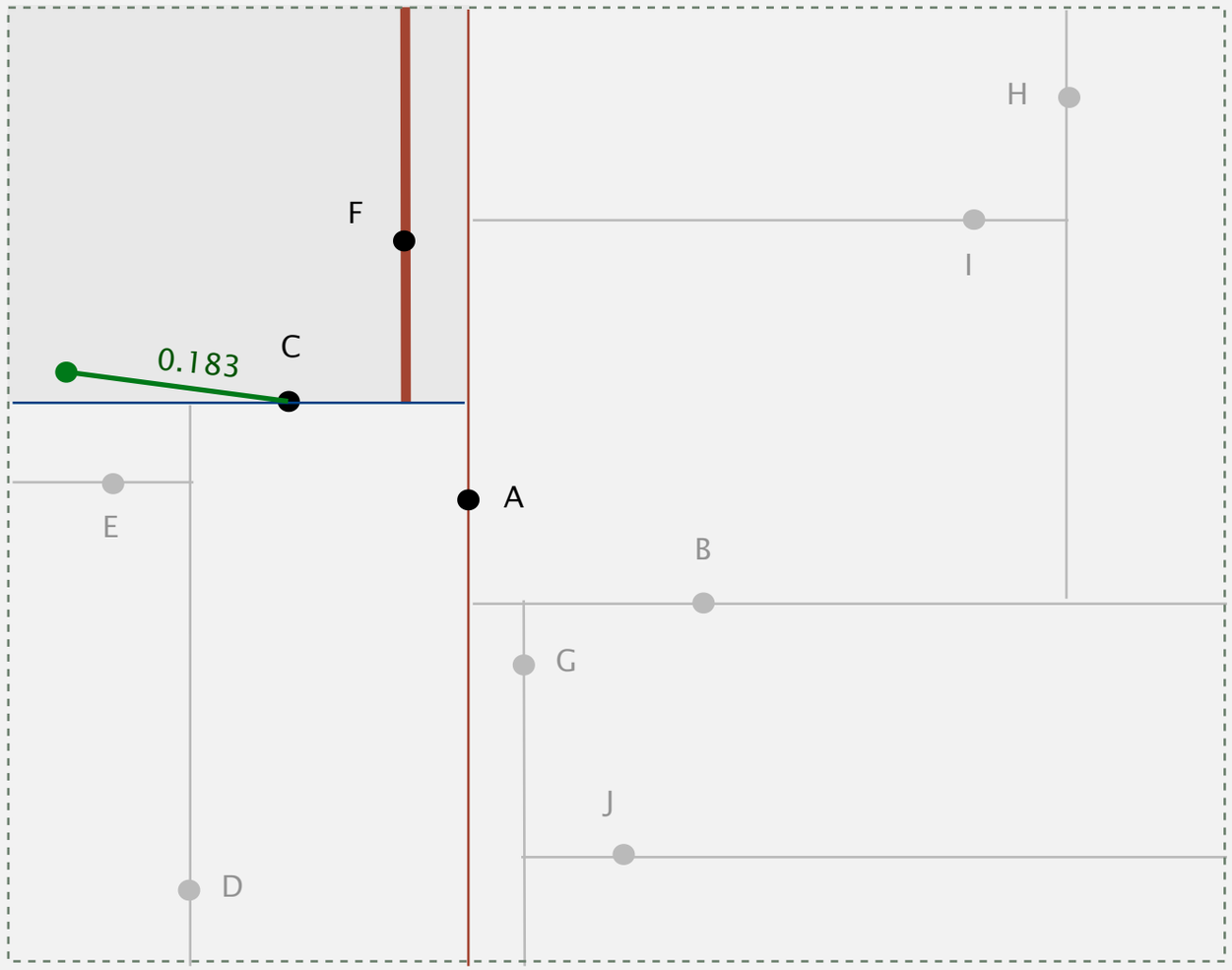


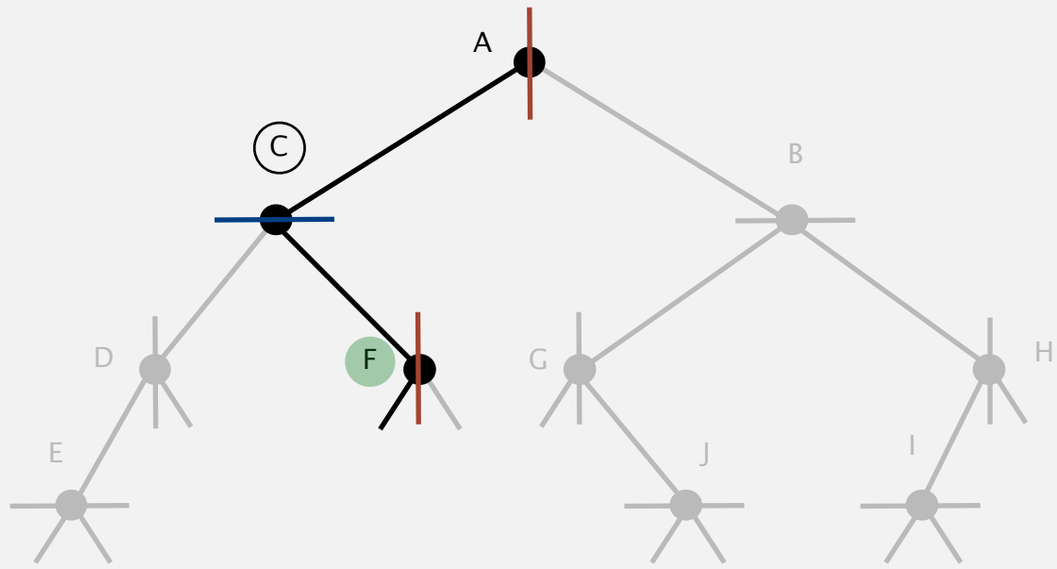
query point is to left of splitting line

search left subtree first

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



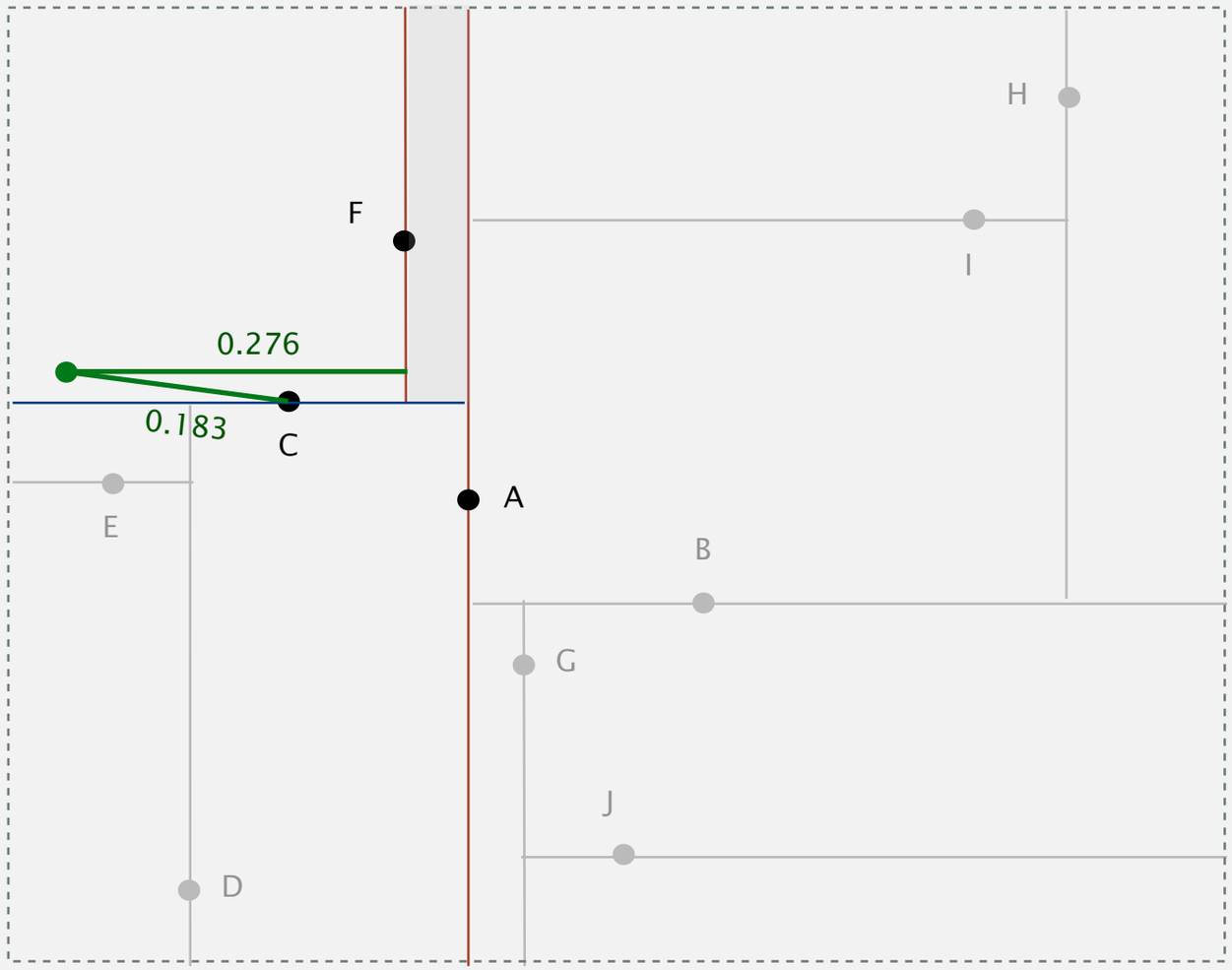


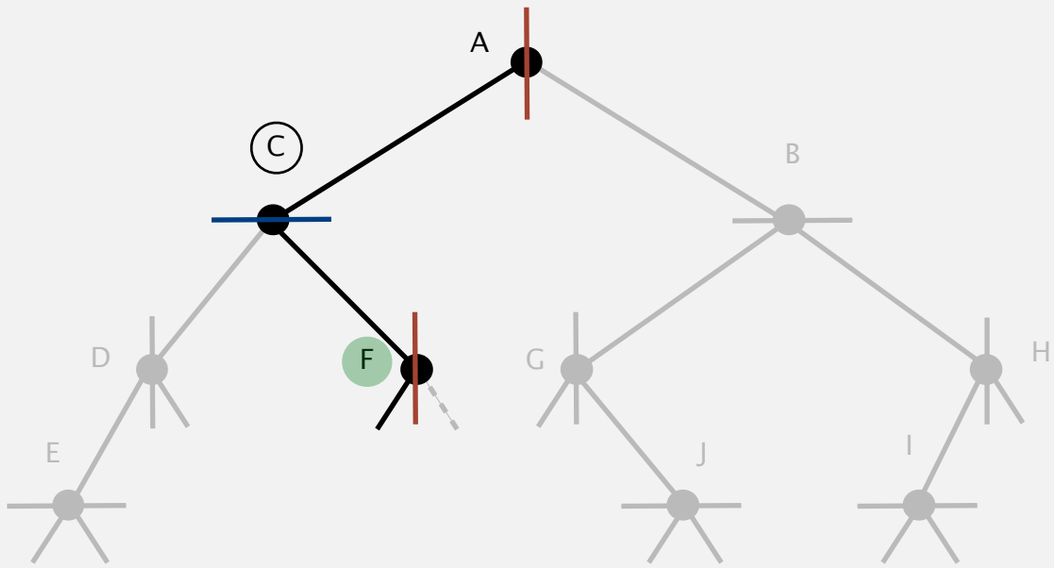
search left subtree

return since empty

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.





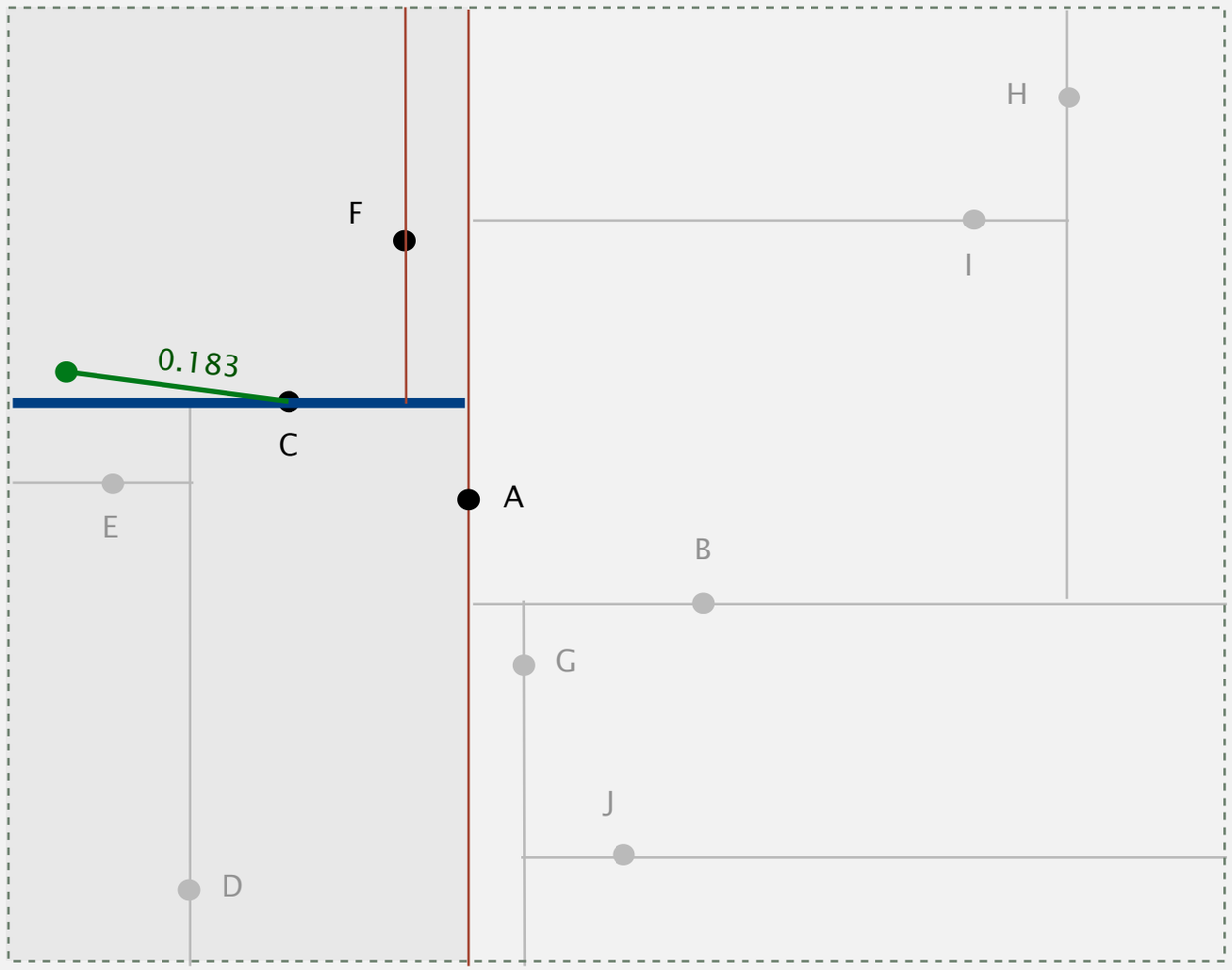
search right subtree

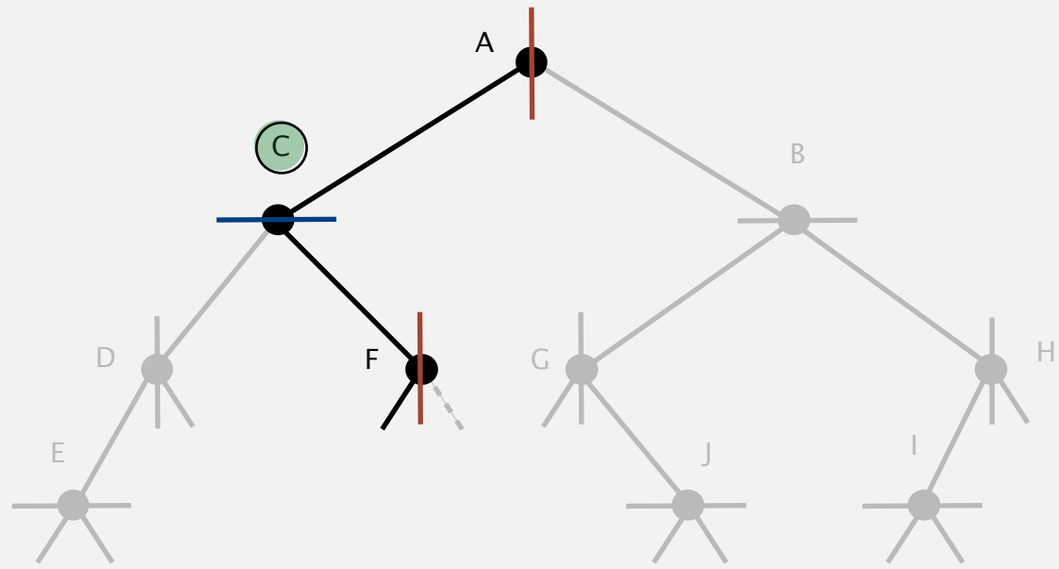
prune since nearest neighbor

can't be here

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



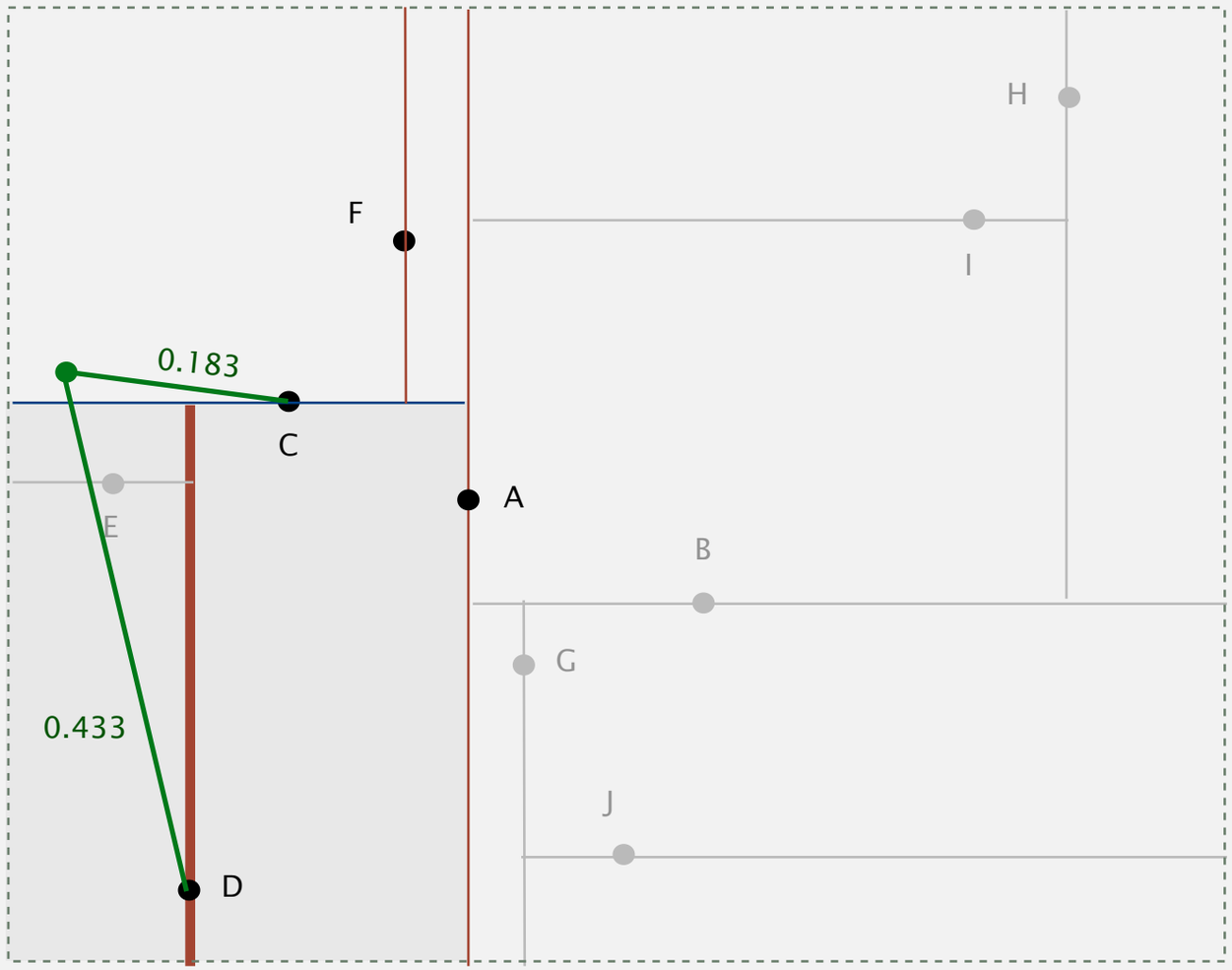


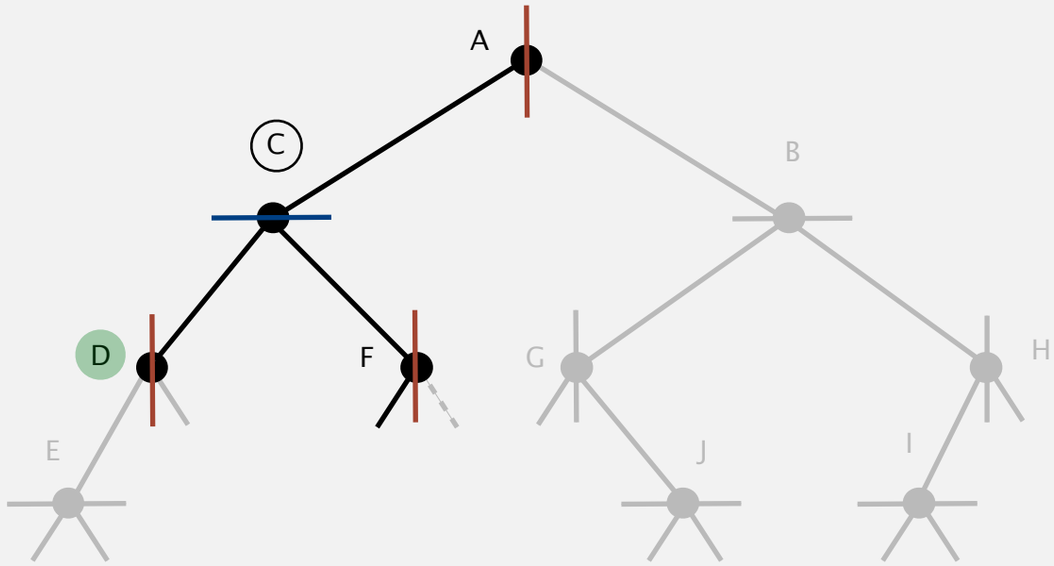
return from function call

search bottom subtree next

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



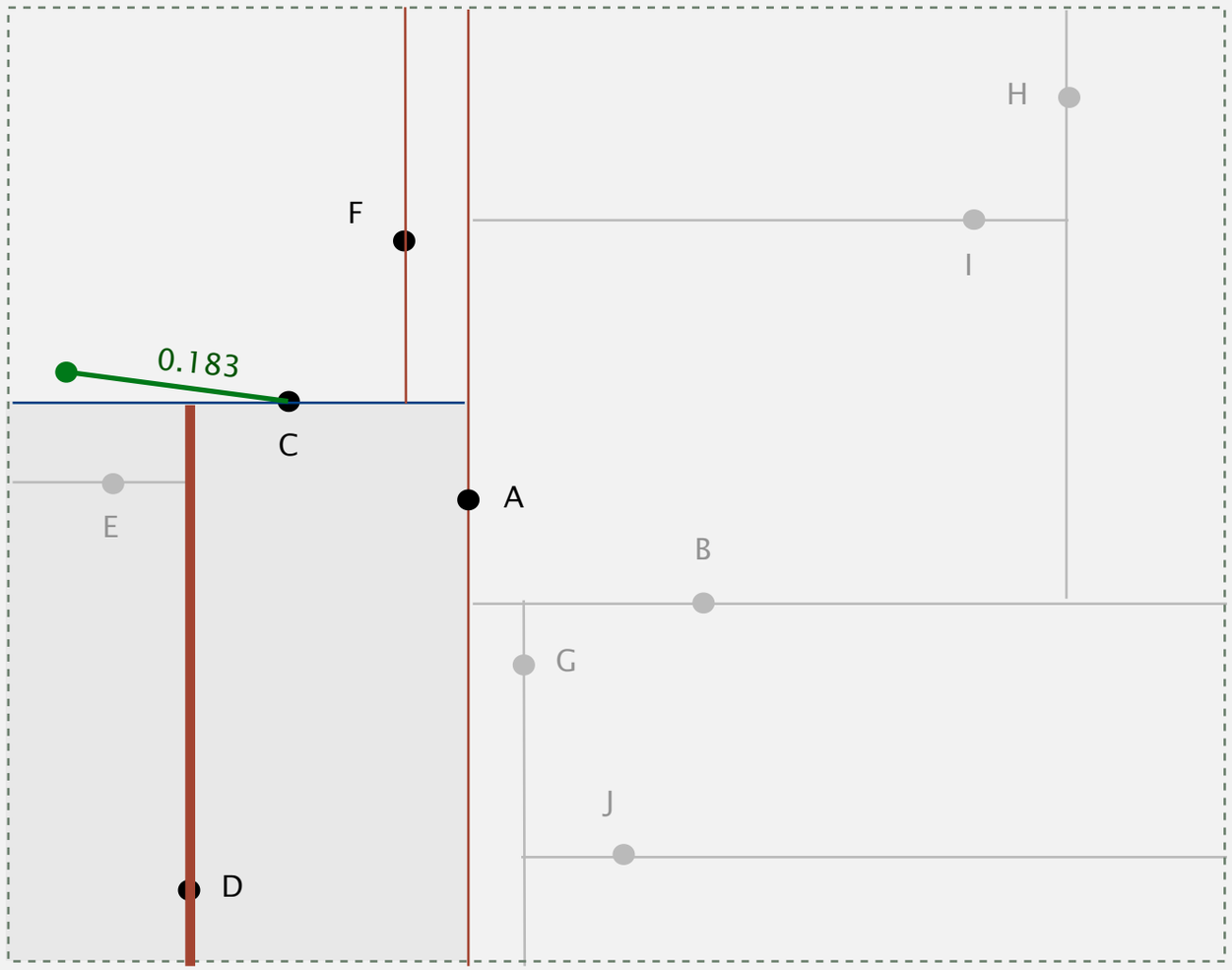


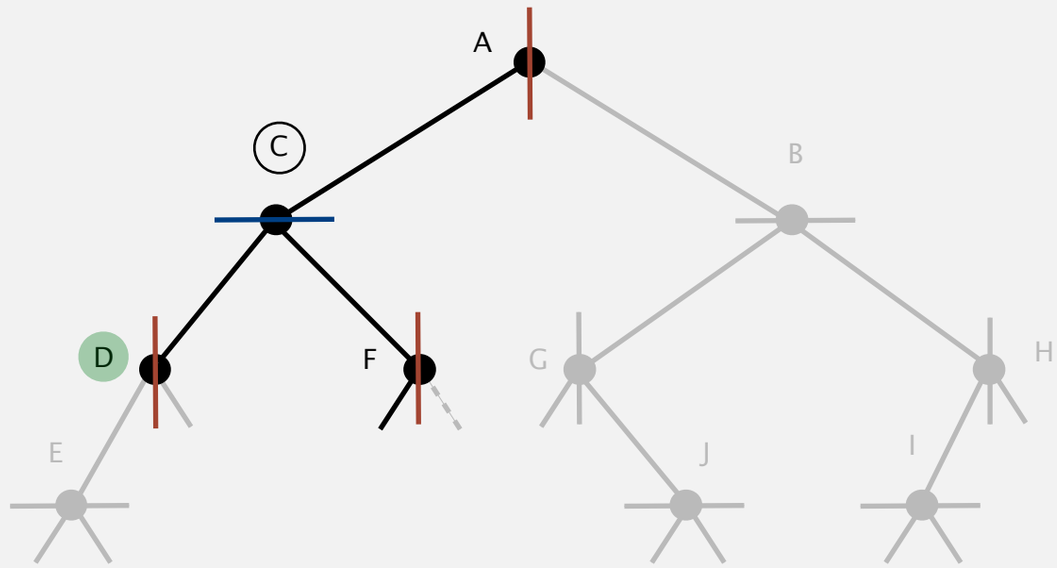
search bottom subtree

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 and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



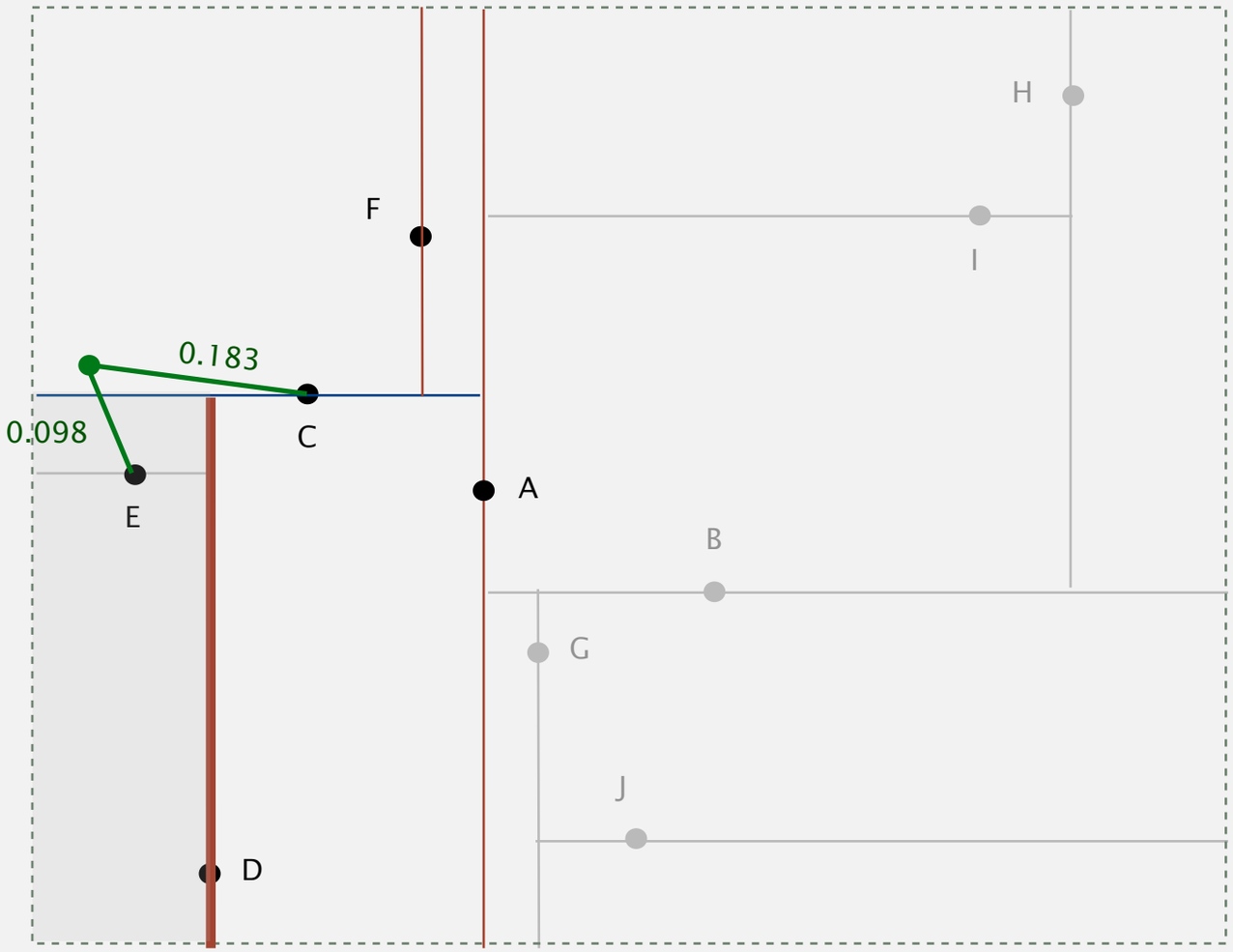


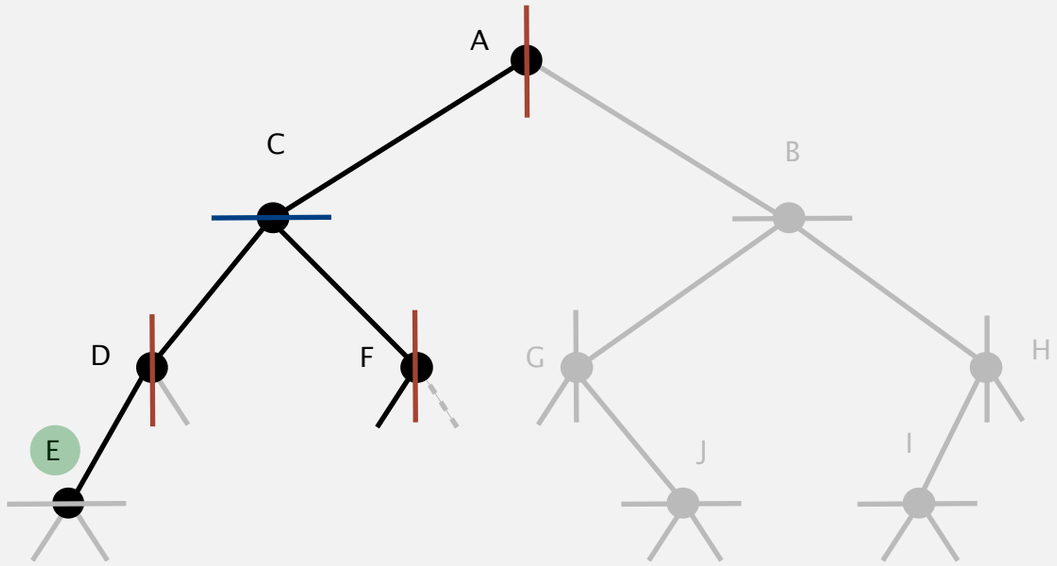
query point is to left of splitting line

search left subtree first

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.





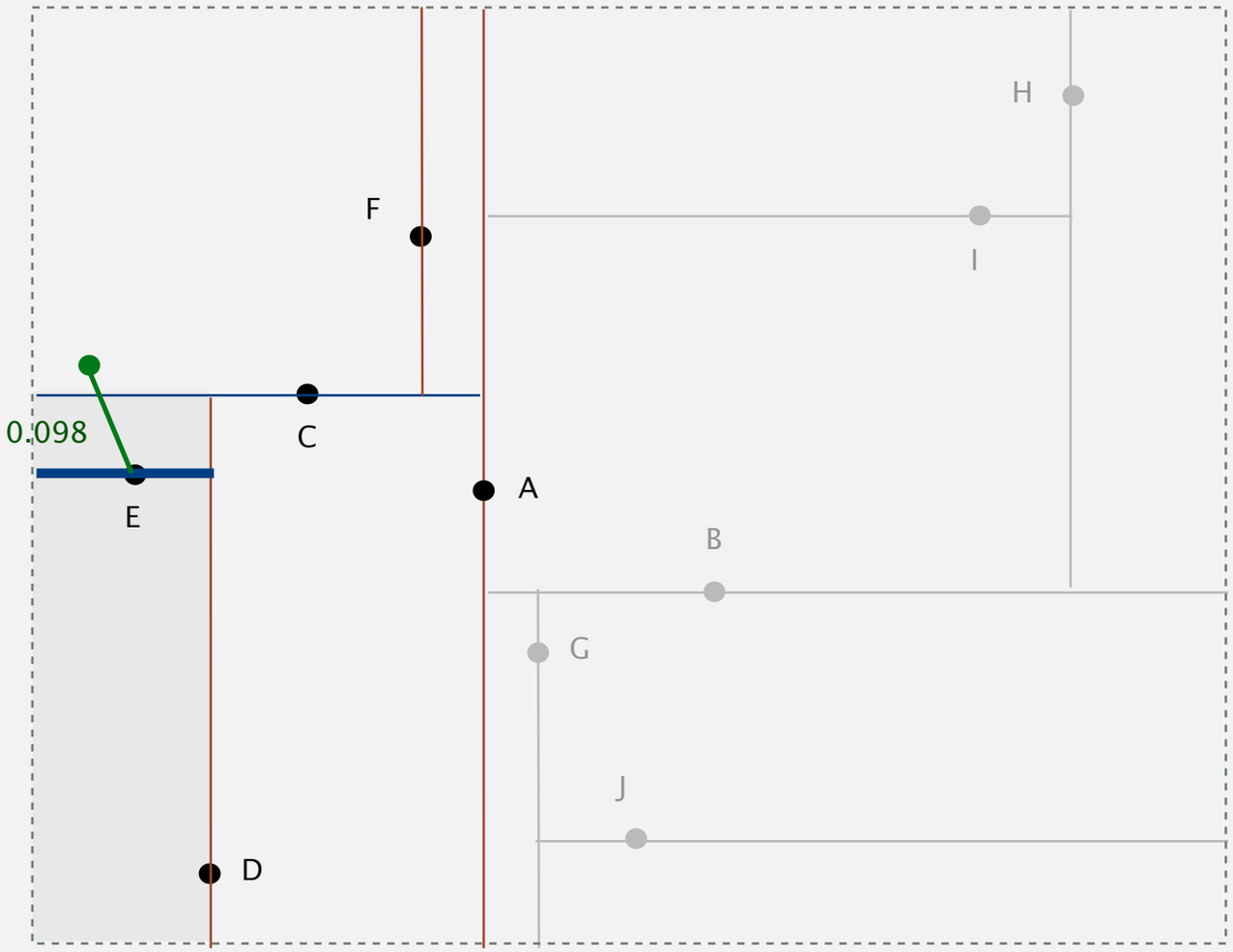
search left subtree

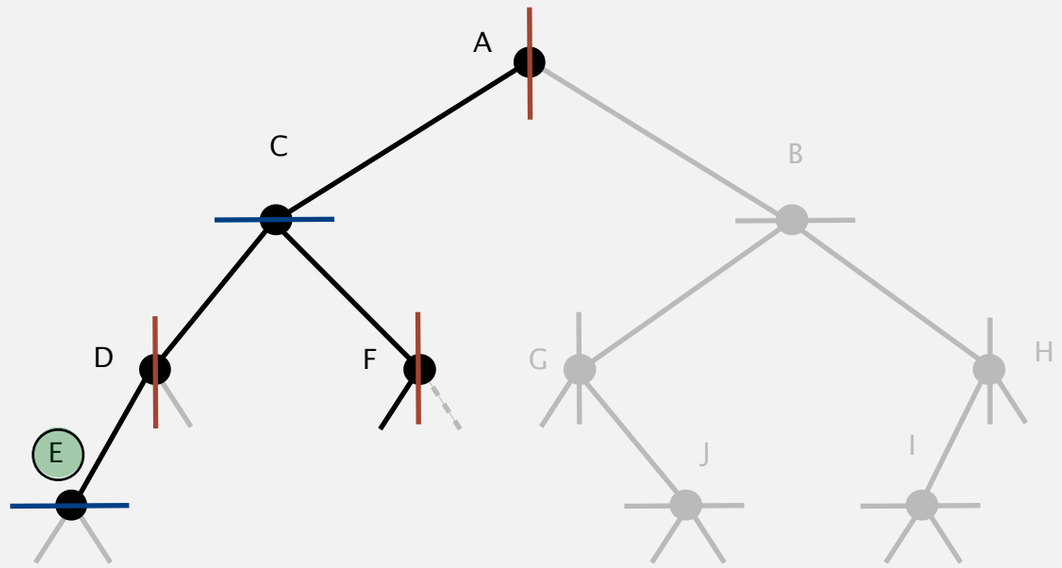
compute distance from query point to E

(update champion)

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



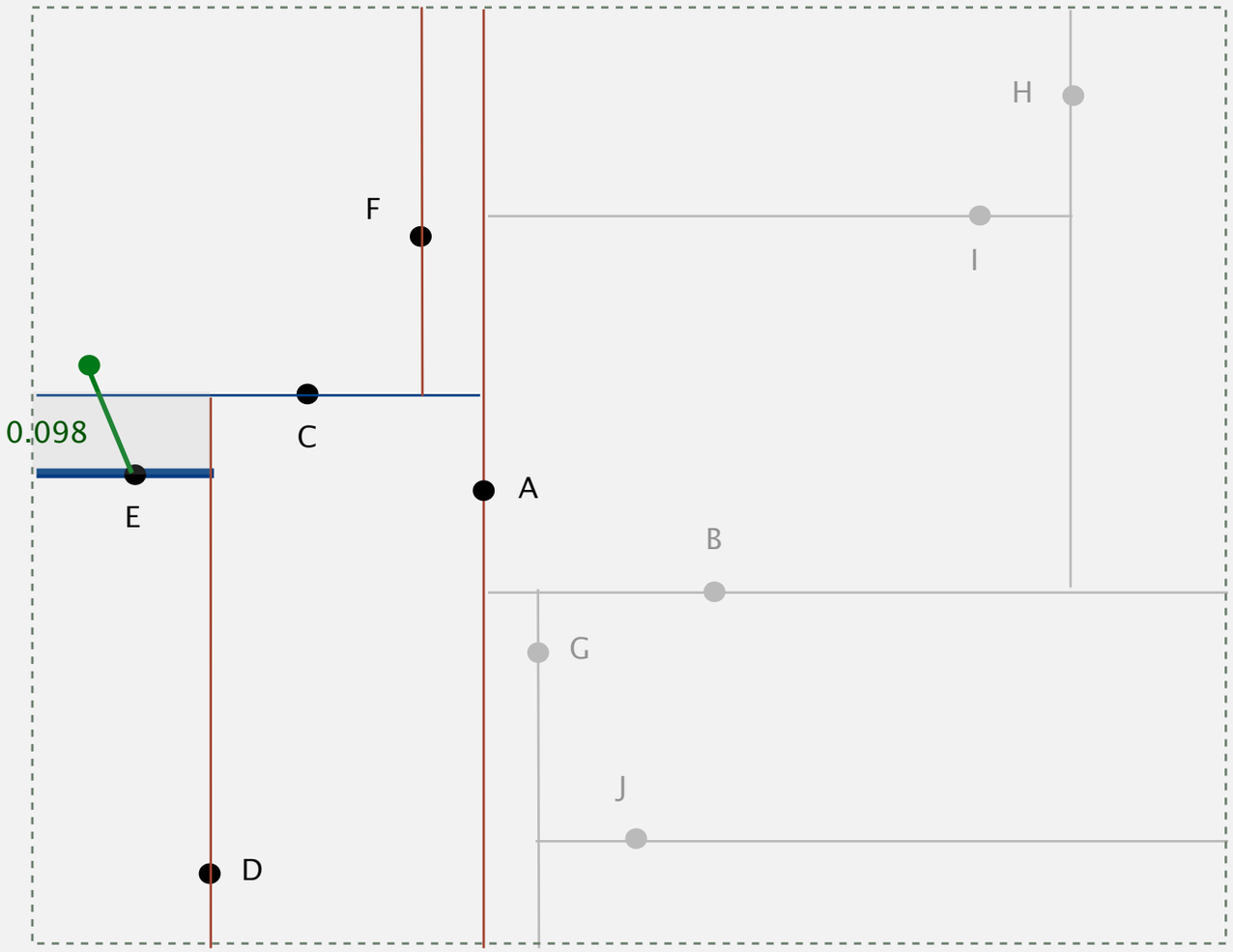


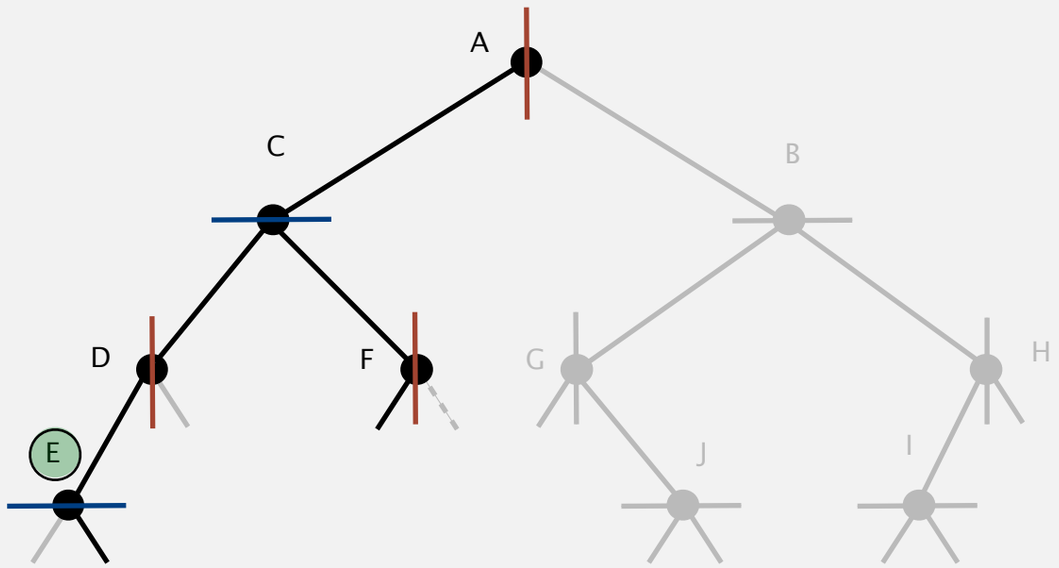
query point is above splitting line

search top subtree first

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



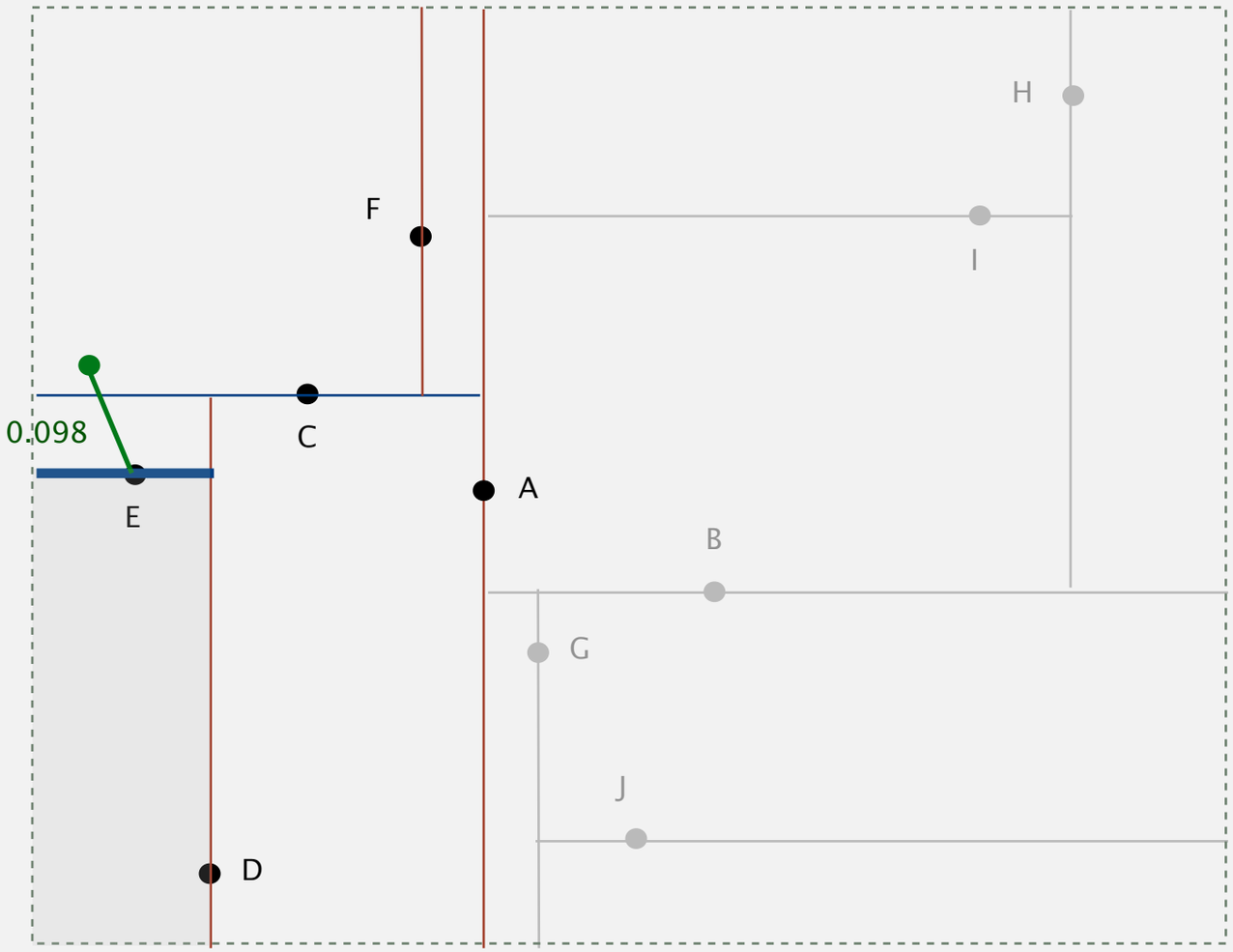


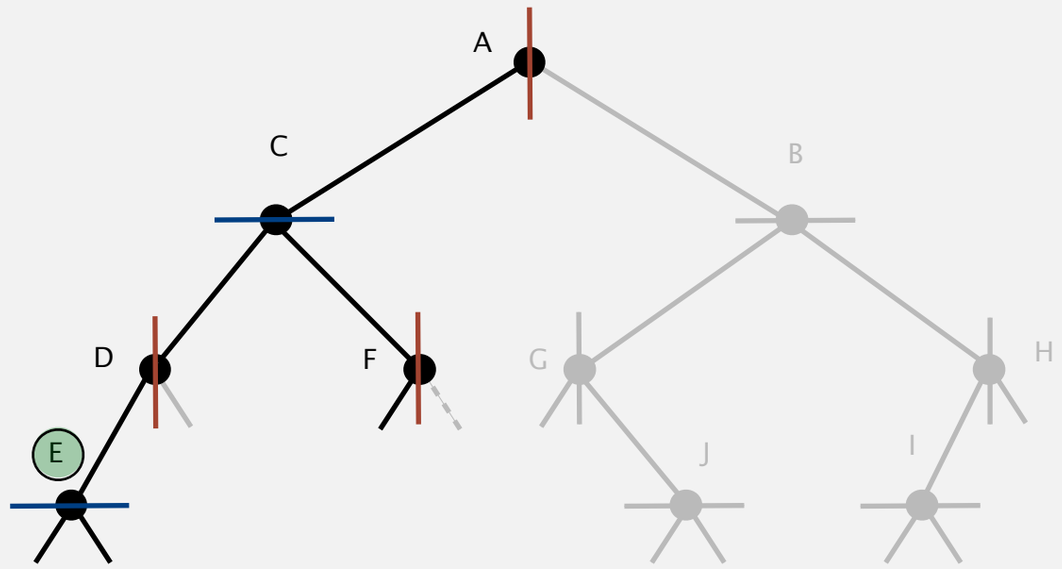
search top subtree

return since empty

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



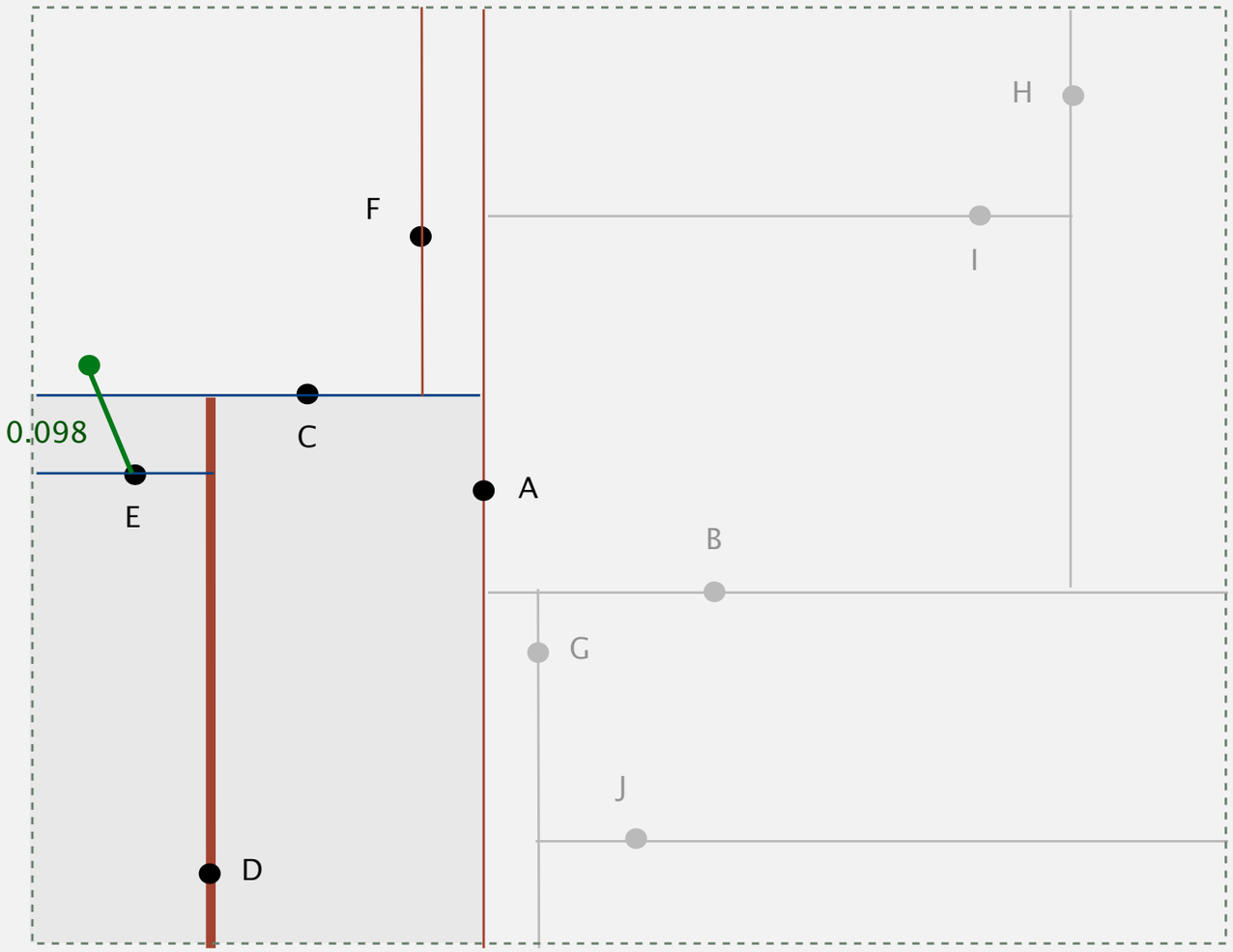


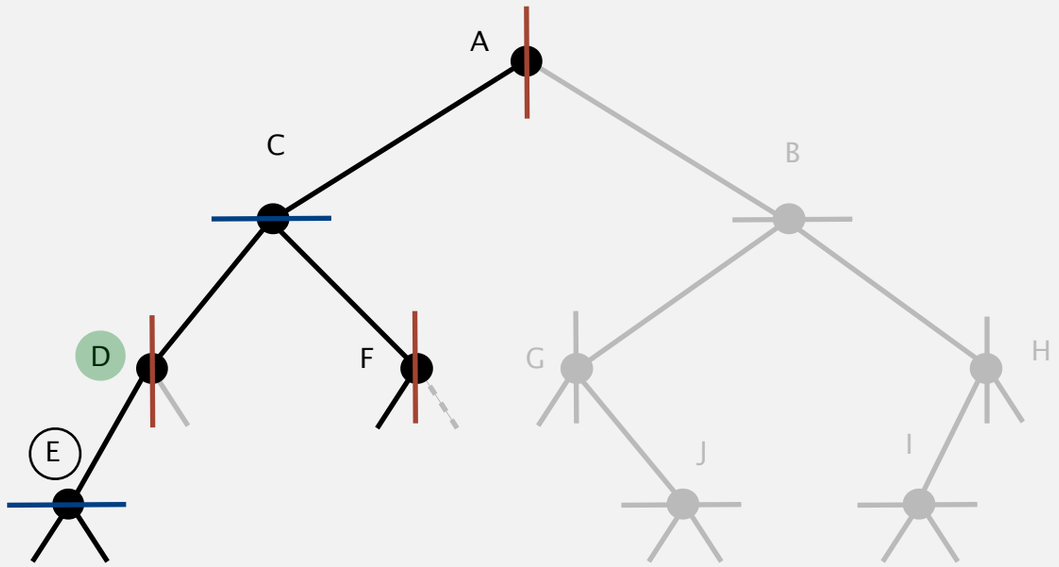
search bottom subtree

return since empty

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.





return from function call

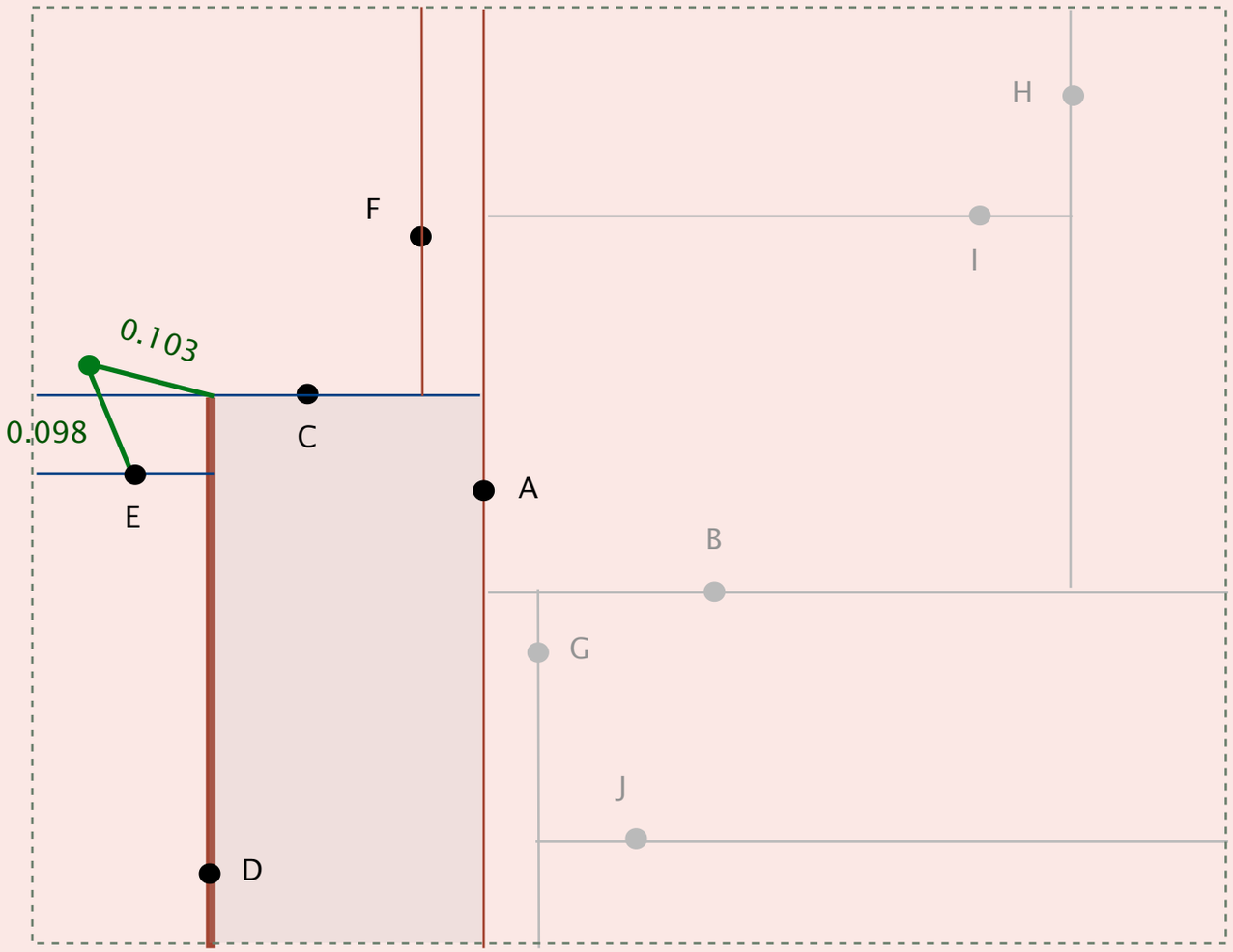
search right subtree next

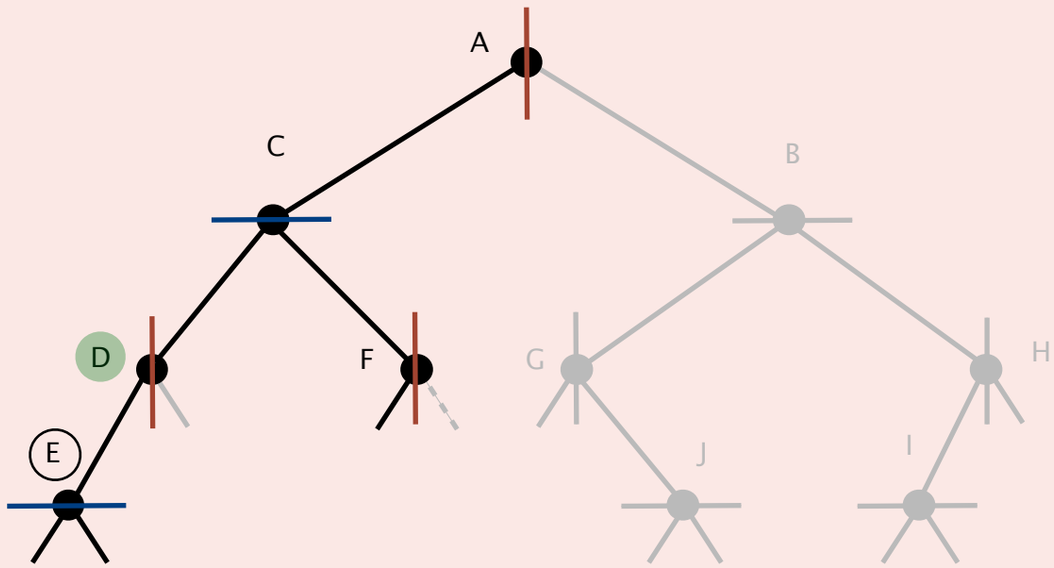
Geometric applications of BSTs: quiz 5

Is it safe to prune right subtree of D?

A. Yes.

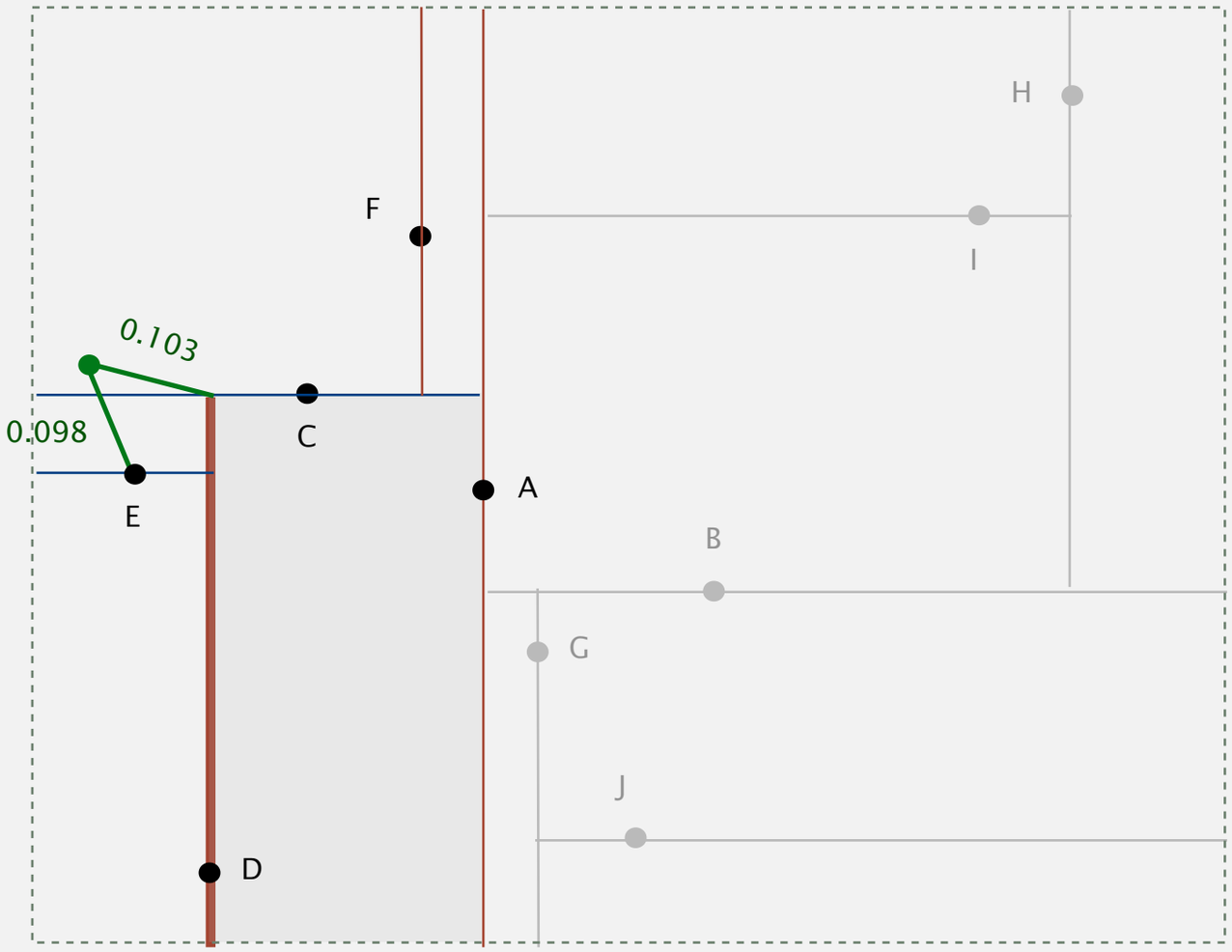
B. No.

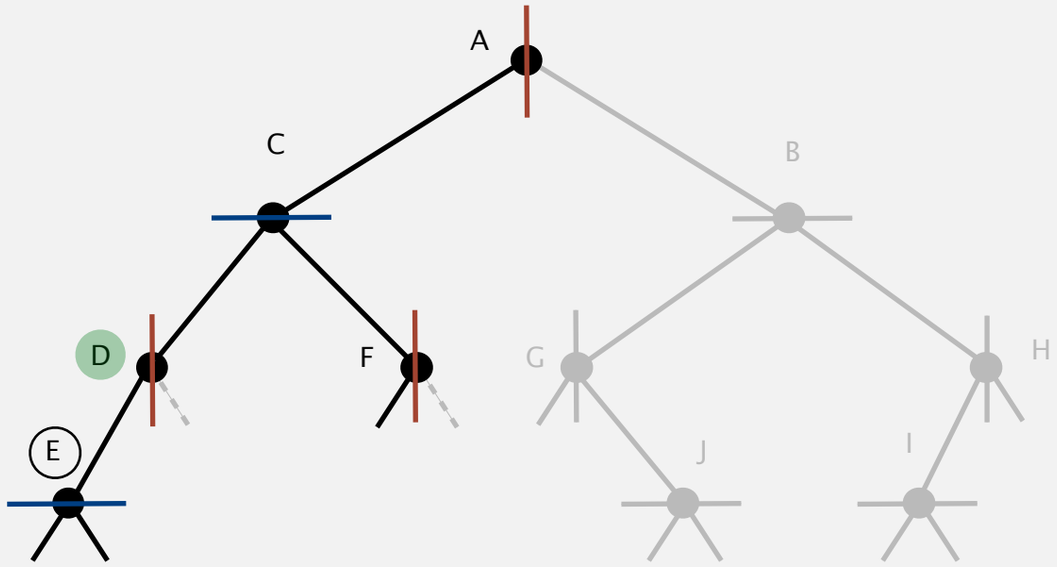




2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.

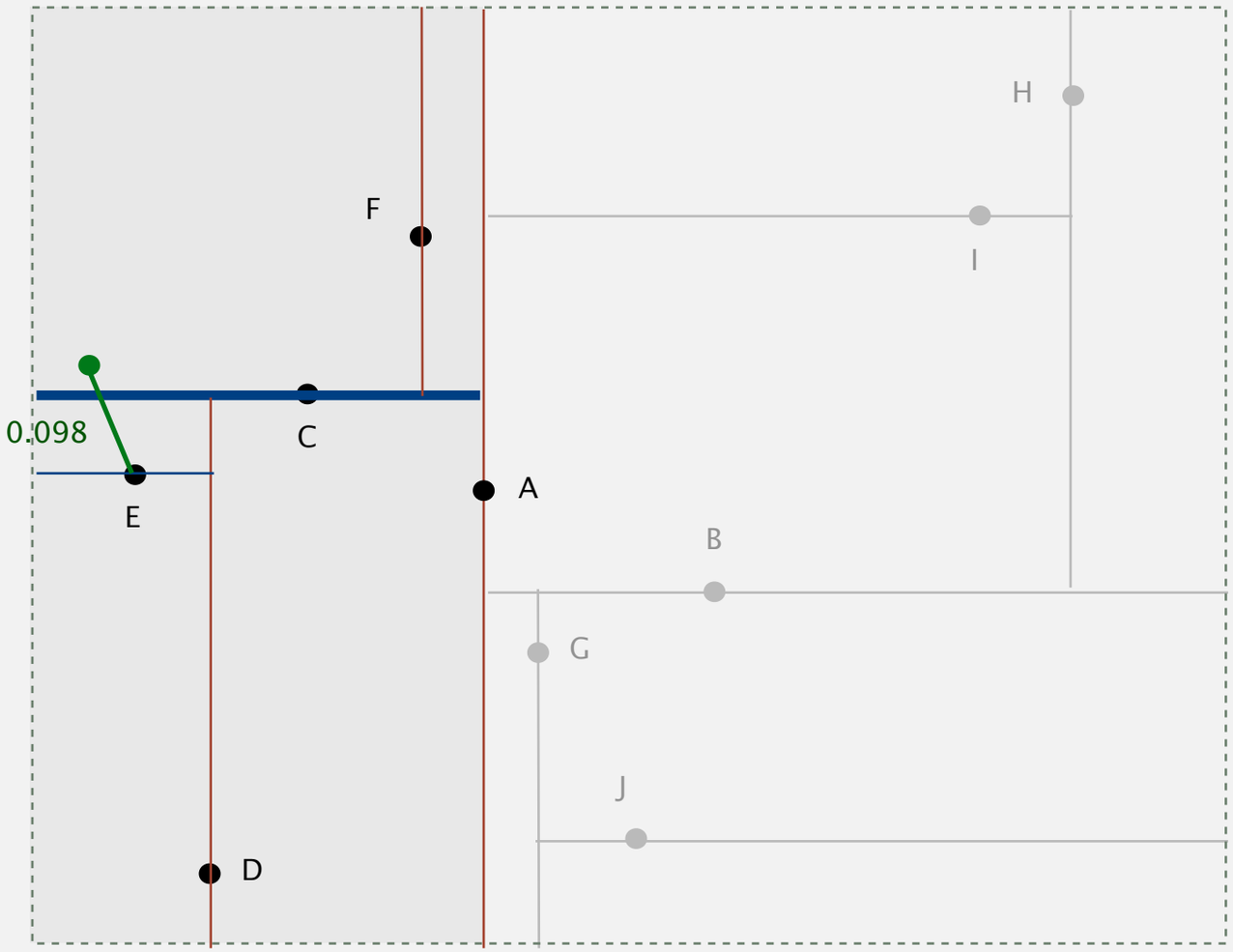


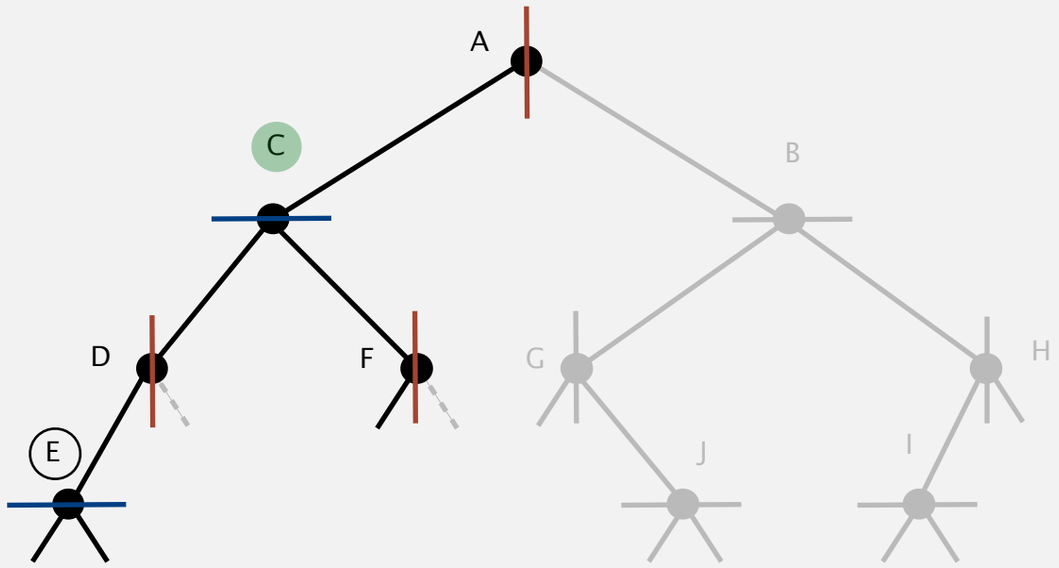


search right subtree
prune since nearest neighbor
can't be here

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.

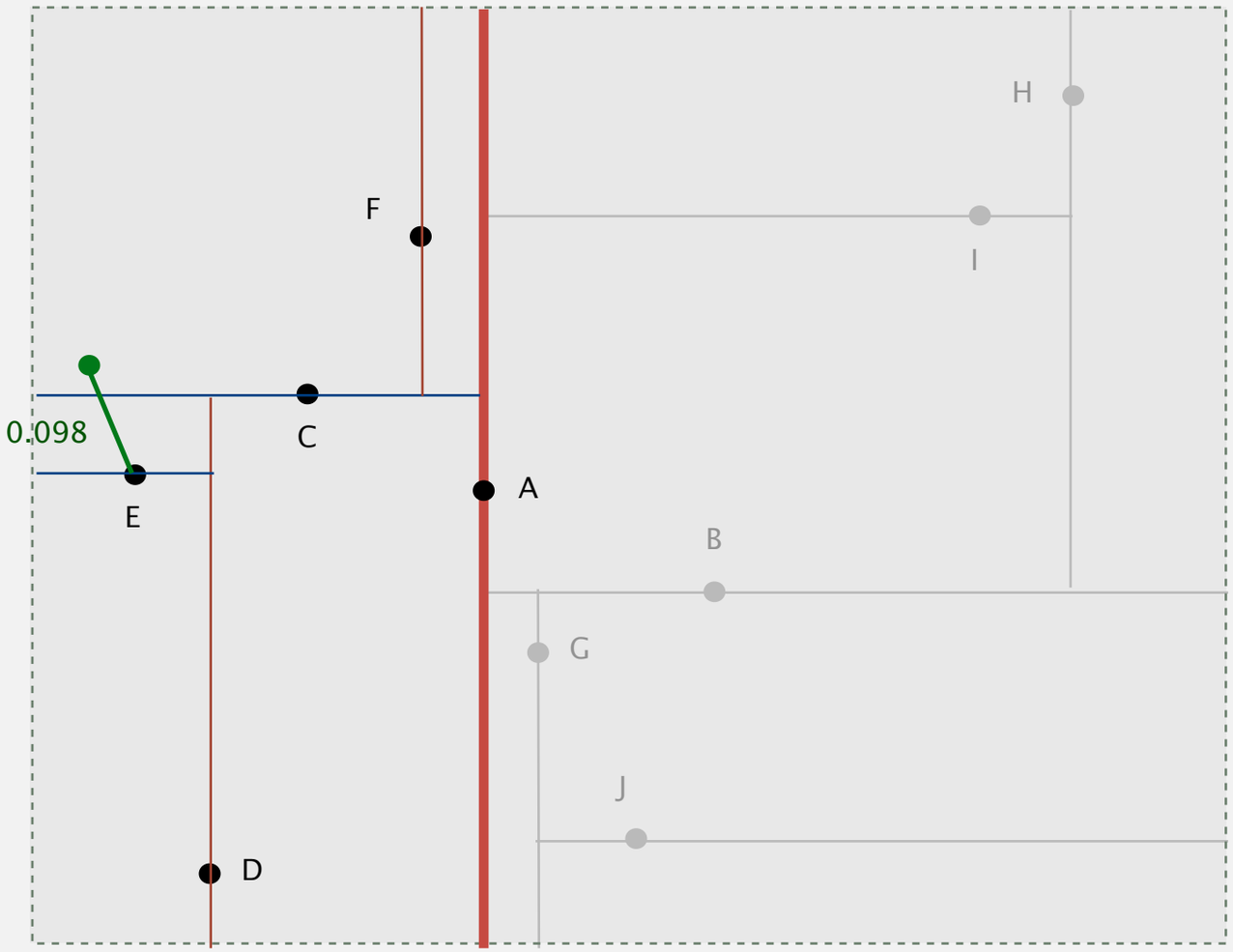


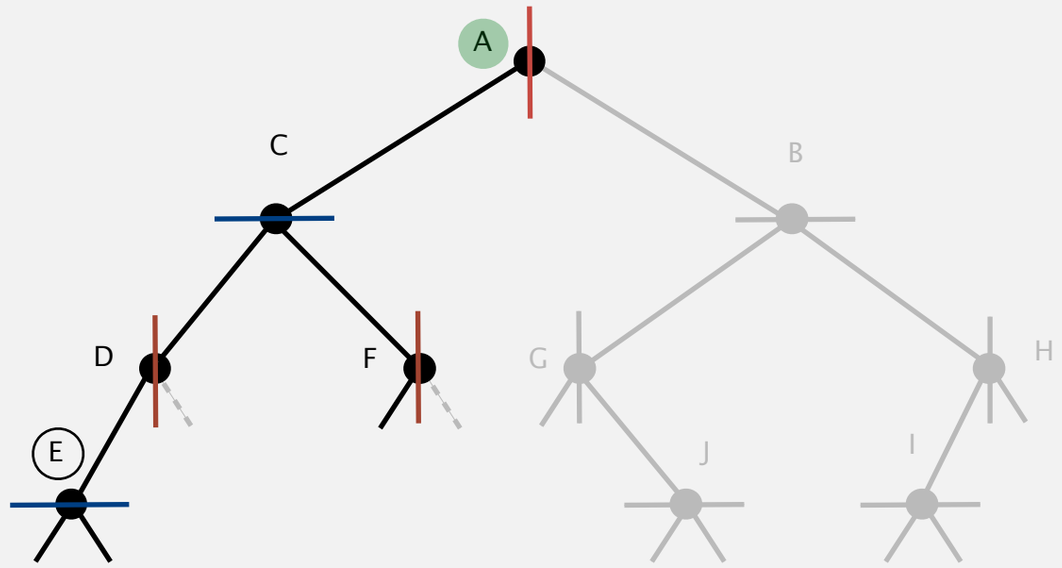


return from function call

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.



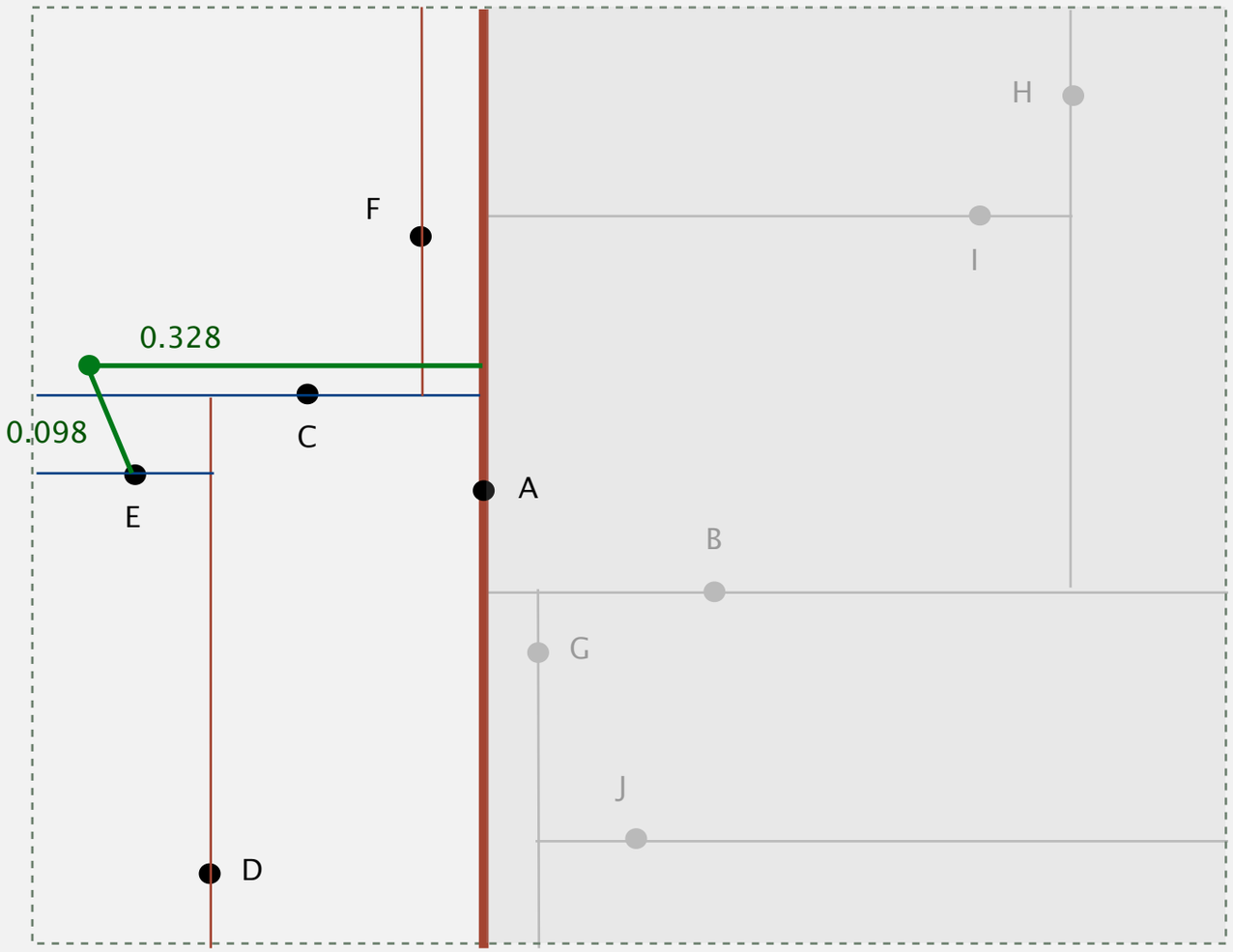


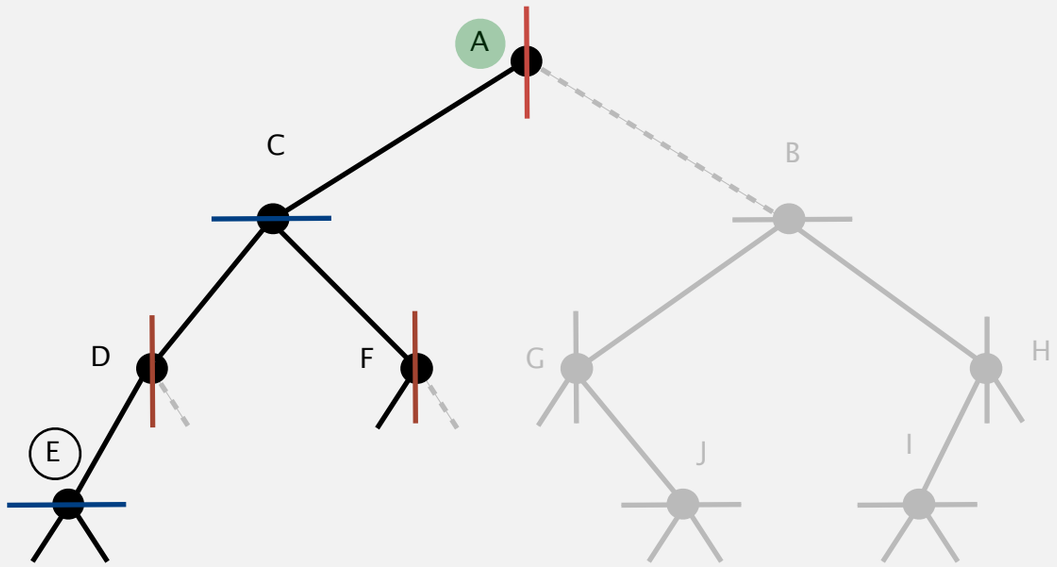
return from function call

search right subtree next

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.





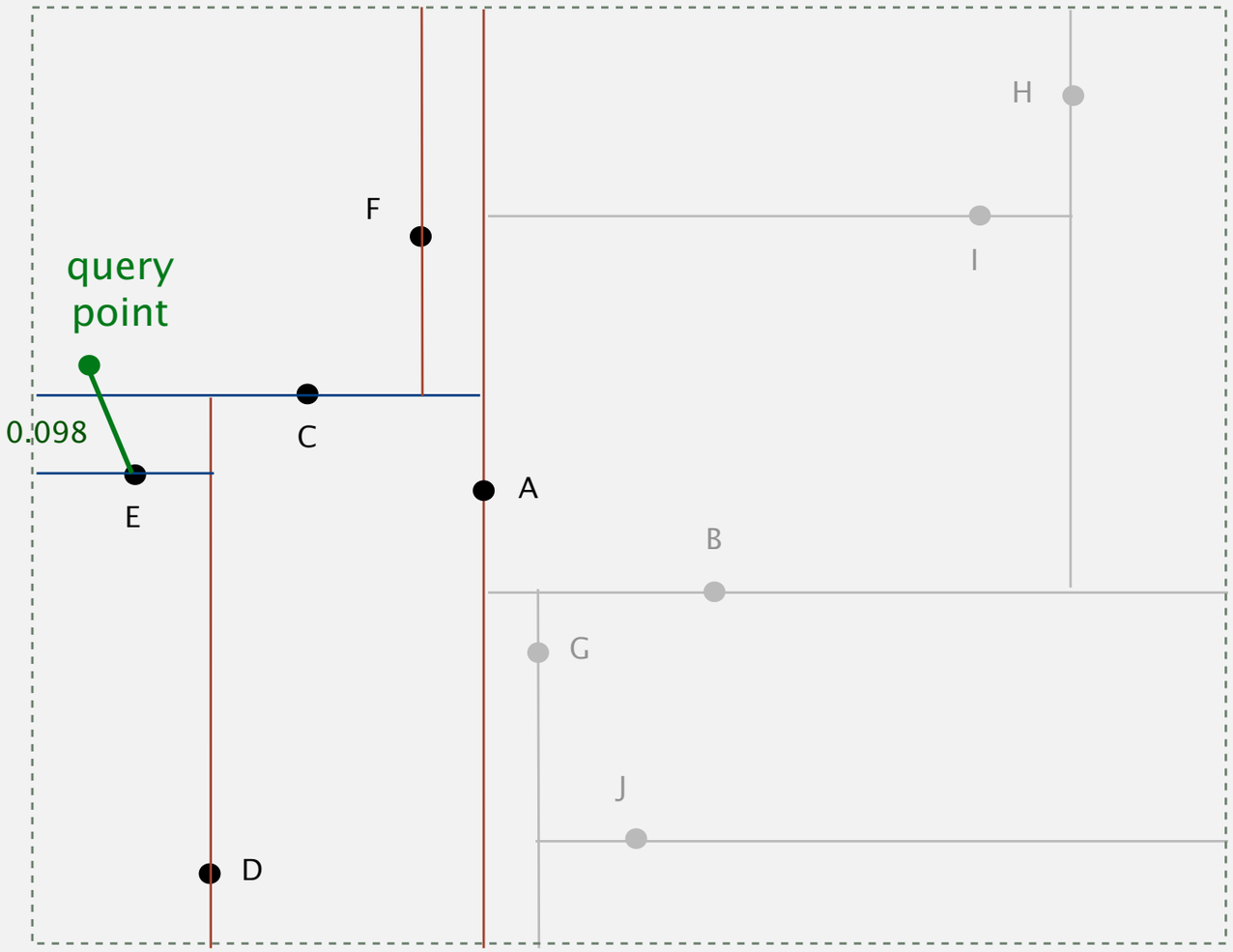
search right subtree

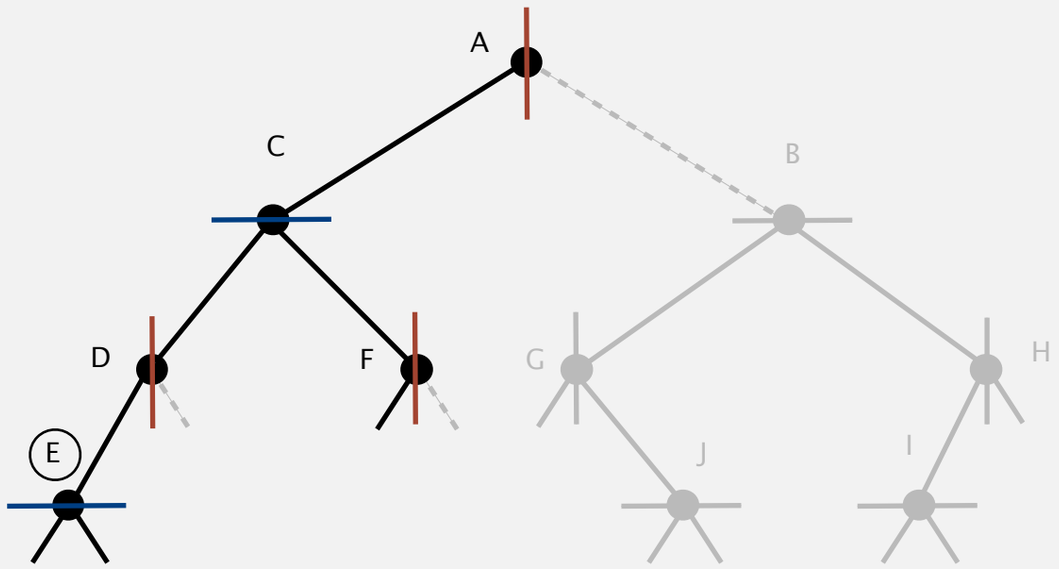
prune since nearest neighbor

can't be here

2d tree demo: nearest neighbor

- Check distance from point in node to query point.
- Recursively search left/bottom and right/top subtrees.
- Optimization 1: prune subtree if it can't contain a closer point.
- Optimization 2: explore subtree toward the query point first.





nearest neighbor = E