2D Tree Demo

- insertion
- range search
- nearest neighbor

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.

A

B

C

A

B

C

D
2d tree demo: insertion

Recursively partition plane into two halfplanes.
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.
Recursively partition plane into two halfplanes.

2d tree demo: insertion
2d tree demo: insertion

Recursively partition plane into two halfplanes.

A
B
C
D
E

A
B
C
D
E

E
D
C
B
A
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.
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.
Recursively partition plane into two halfplanes.
2d tree demo: insertion

Recursively partition plane into two halfplanes.
Recursively partition plane into two halfplanes.
2d tree demo: insertion

Recursively partition plane into two halfplanes.
Recursively partition plane into two halfplanes.
2d Tree Demo

- insertion
- range search
- nearest neighbor
Goal. Find all points in a query rectangle.
Goal. Find all points in a query rectangle.

(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.
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.

![Diagram showing a 2d tree with a range search demo.](image-url)
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
**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.
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.
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.

Geometric applications of BSTs: quiz 3
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
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.

**2d tree demo: range search**

search left subtree
check if query rectangle contains point E
(search hit)
**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.
**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.
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.
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.
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.
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.
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
**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.
**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.

2d tree demo: range search

search left subtree
stop since empty
**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.
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.
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.
**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.
2D TREE DEMO

- insertion
- range search
- nearest neighbor
Goal. Find closest point to query point.
Goal. Find closest point to query point.
**Goal.** Find closest point to query point.
2d tree demo: nearest neighbor

- Check distance from point in node 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 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.
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.
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
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.
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
• 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.
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.
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.
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.
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.
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.
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.
• 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.
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.
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.
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