Kd Tree Demo

- insertion
- range search
- nearest neighbor search

click to begin demo
- insertion
- range search
- nearest neighbor search
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

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

Insertion in a 2d tree
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

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

Insertion in a 2d tree
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

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

Insertion in a 2d tree
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

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

Insertion in a 2d tree
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
Insertion in a 2d tree

Recursively partition plane into two halfplanes.
- insertion
- range search
- nearest neighbor search
Range search in a 2d tree

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

find all points in 2d tree that are contained in green query rectangle
Range search in a 2d tree

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

search root node
check if query rectangle contains point 1
Range search in a 2d tree

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

query rectangle to left of splitting line
search only in left subtree
Range search in a 2d tree

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

search left subtree
check if query rectangle contains point 3
Range search in a 2d tree

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

query rectangle intersects splitting line
search bottom and top subtrees
Range search in a 2d tree

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

search left subtree
check if query rectangle contains point 4
Range search in a 2d tree

- Check if point in node lies in given rectangle.
- Recursively search left/bottom subdivision (if any could fall in rectangle).
- Recursively search right/top subdivision (if any could fall in rectangle).
Range search in a 2d tree

- Check if point in node lies in given rectangle.
- Recursively search left/bottom subdivision (if any could fall in rectangle).
- Recursively search right/top subdivision (if any could fall in rectangle).
Range search in a 2d tree

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

query rectangle intersects splitting line
search bottom and top subtrees
Range search in a 2d tree

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

search bottom subtree
stop since empty
Range search in a 2d tree

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

search top subtree
stop since empty
Range search in a 2d tree

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

return from function call
Range search in a 2d tree

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

return from function call
Range search in a 2d tree

- Check if point in node lies in given rectangle.
- Recursively search left/bottom subdivision (if any could fall in rectangle).
- Recursively search right/top subdivision (if any could fall in rectangle).
Range search in a 2d tree

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

search top subtree
check if query rectangle contains point 6
Range search in a 2d tree

- Check if point in node lies in given rectangle.
- Recursively search left/bottom subdivision (if any could fall in rectangle).
- Recursively search right/top subdivision (if any could fall in rectangle).
Range search in a 2d tree

- Check if point in node lies in given rectangle.
- Recursively search left/bottom subdivision (if any could fall in rectangle).
- Recursively search right/top subdivision (if any could fall in rectangle).
Range search in a 2d tree

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

```
return from function call
```
Range search in a 2d tree

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

return from function call
• Check if point in node lies in given rectangle.
• Recursively search left/bottom subdivision (if any could fall in rectangle).
• Recursively search right/top subdivision (if any could fall in rectangle).
Range search in a 2d tree

- Check if point in node lies in given rectangle.
- Recursively search left/bottom subdivision (if any could fall in rectangle).
- Recursively search right/top subdivision (if any could fall in rectangle).
• insertion
• range search
• nearest neighbor search
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

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

search root node
compute distance from query point to 1
(update champion nearest neighbor)
Nearest neighbor search in a 2d tree

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

query point is to the left of splitting line
search left subtree first
Nearest neighbor search in a 2d tree

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

search left subtree
compute distance from query point to 3
(update champion)
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

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

search top subtree
compute distance from query point to 6
Nearest neighbor search in a 2d tree

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

query point is to left of splitting line
search left subtree first
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

• Check distance from point in node to query point.
• Recursively search left/bottom subdivision (if it could contain a closer point).
• Recursively search right/top subdivision (if it could contain a closer point).
• Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

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

query point is above splitting line
search top subtree first
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

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

search right subtree
prune since nearest neighbor
can't be in subdivision
Nearest neighbor search in a 2d tree

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

return from function call
Nearest neighbor search in a 2d tree

- Check distance from point in node to query point.
- Recursively search left/bottom subdivision (if it could contain a closer point).
- Recursively search right/top subdivision (if it could contain a closer point).
- Organize recursive method so that it begins by searching for query point.
Nearest neighbor search in a 2d tree

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

search right subtree
prune since nearest neighbor can't be in subdivision
Nearest neighbor search in a 2d tree

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

nearest neighbor = 5