

Problem 1 The idea behind the range query on binary search tree is traversing the tree inorder while the node keys remain in range and returning every node within the range. Therefore the algorithm follows the steps below:

- Recursively call the left sub-tree if the root key is greater or equal to the lower limit.
- If the root key is within range keep the key
- Recursively call the right sub-tree.

The pseudo-code:

```
range_query(root, lower_limit, upper_limit, keys_found){
    if(root->key >= lower_limit && root->left != NULL)
        range_query(root->left, lower_limit, upper_limit, keys_found)

    if(root->key >= lower_limit && root->key <= upper_limit)
        keys_found.add(root->key)

    if(root->right != NULL)
        range_query(root->right, lower_limit, upper_limit, keys_found)

    return
}
```

We can see that in general we have a complexity of $O(h)$ for the recursions and complexity of $O(k)$ for the keys found.

Therefore the total complexity is $O(h + k)$.

Problem 2 We know that the convex combination of two points describe all the points in the line segment between those two points. Whereas the affine combination of two

points describes all the point on the line through those two points.

Convex Combination of two points $\mathbf{x}_1, \mathbf{x}_2$:

$$\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2, \lambda \in [0, 1]$$

Affine Combination of two points $\mathbf{x}_1, \mathbf{x}_2$:

$$\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2, \lambda \in \mathcal{R}$$

Therefore for the pairs of endpoints $(\mathbf{p}_1, \mathbf{p}_2)$ and $(\mathbf{q}_1, \mathbf{q}_2)$, we need to calculate the λ_1 and λ_2 for which the affine combinations of $(\mathbf{p}_1, \mathbf{p}_2)$ and $(\mathbf{q}_1, \mathbf{q}_2)$ respectively, give the same point (the intersection). If $\lambda_1, \lambda_2 \in [0, 1]$, the intersection belongs in the line segments and thus they intersect.

Calculation of λ_1, λ_2 :

$$\text{Let } \mathbf{p}_1 = \begin{bmatrix} p_1(1) \\ p_1(2) \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} p_2(1) \\ p_2(2) \end{bmatrix}, \mathbf{q}_1 = \begin{bmatrix} q_1(1) \\ q_1(2) \end{bmatrix}, \mathbf{q}_2 = \begin{bmatrix} q_2(1) \\ q_2(2) \end{bmatrix} \text{ assuming that } \mathbf{p}_1 \neq \mathbf{p}_2, \mathbf{q}_1 \neq \mathbf{q}_2$$

The points given by the affine combination of the pair $(\mathbf{p}_1, \mathbf{p}_2)$:

$$\mathbf{p} = \lambda_1 \mathbf{p}_1 + (1 - \lambda_1) \mathbf{p}_2 = \lambda_1 (\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{p}_2$$

The points given by the affine combination of the pair $(\mathbf{q}_1, \mathbf{q}_2)$:

$$\mathbf{q} = \lambda_2 \mathbf{q}_1 + (1 - \lambda_2) \mathbf{q}_2 = \lambda_2 (\mathbf{q}_1 - \mathbf{q}_2) + \mathbf{q}_2$$

First we check if the vectors defined by the endpoints are parallel, using the cross product:

$$|(\mathbf{q}_2 - \mathbf{q}_1) \times (\mathbf{p}_2 - \mathbf{p}_1)| = (q_2(1) - q_1(1))(p_2(2) - p_1(2)) - (q_2(2) - q_1(2))(p_2(1) - p_1(1))$$

- If $(q_2(1) - q_1(1))(p_2(2) - p_1(2)) - (q_2(2) - q_1(2))(p_2(1) - p_1(1)) = 0$ then the line segments are parallel. Then the two line segments intersect only if one or both of the endpoints of one line segment belong in the other.

Which means that one of the following equalities must be true for some $\lambda \in$

$[0, 1]$:

$$\begin{aligned}
\mathbf{p}_1 &= \lambda_1(\mathbf{q}_1 - \mathbf{q}_2) + \mathbf{q}_2 \iff \begin{cases} p_1(1) = \lambda_{1,x}(q_1(1) - q_2(1)) + q_2(1) \\ p_1(2) = \lambda_{1,y}(q_1(2) - q_2(2)) + q_2(2) \end{cases} \\
\mathbf{p}_2 &= \lambda_2(\mathbf{q}_1 - \mathbf{q}_2) + \mathbf{q}_2 \iff \begin{cases} p_2(1) = \lambda_{2,x}(q_1(1) - q_2(1)) + q_2(1) \\ p_2(2) = \lambda_{2,y}(q_1(2) - q_2(2)) + q_2(2) \end{cases} \\
\mathbf{q}_1 &= \lambda_3(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{p}_2 \iff \begin{cases} q_1(1) = \lambda_{3,x}(p_1(1) - p_2(1)) + p_2(1) \\ q_1(2) = \lambda_{3,y}(p_1(2) - p_2(2)) + p_2(2) \end{cases} \\
\mathbf{q}_2 &= \lambda_4(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{p}_2 \iff \begin{cases} q_2(1) = \lambda_{4,x}(p_1(1) - p_2(1)) + p_2(1) \\ q_2(2) = \lambda_{4,y}(p_1(2) - p_2(2)) + p_2(2) \end{cases}
\end{aligned}$$

We can see that if $q_1(1) = q_2(1)$ or $q_1(2) = q_2(2)$ then it must be true that $q_1(1) = q_2(1) = p_1(1) = p_2(1)$ and $q_1(2) = q_2(2) = p_1(2) = p_2(2)$ respectively for the line segments to possibly intersect.

Also it must be true that $\lambda_x = \lambda_y$ for the line segments to be co-linear and to possibly intersect.

Then if any $\lambda \in [0, 1]$, there is an intersection and the algorithm returns TRUE. In the other cases it returns FALSE.

- If $(q_2(1) - q_1(1))(p_2(2) - q_1(2)) - (q_2(2) - q_1(2))(p_2(1) - q_1(1)) \neq 0$.

The intersection is at the point $\mathbf{p} = \mathbf{q}$, so

$$\mathbf{p} = \mathbf{q} \iff$$

$$\lambda_1(\mathbf{p}_1 - \mathbf{p}_2) + \mathbf{p}_2 = \lambda_2(\mathbf{q}_1 - \mathbf{q}_2) + \mathbf{q}_2 \iff$$

$$\begin{aligned}
\lambda_1 \left(\begin{bmatrix} p_1(1) \\ p_1(2) \end{bmatrix} - \begin{bmatrix} p_2(1) \\ p_2(2) \end{bmatrix} \right) + \begin{bmatrix} p_2(1) \\ p_2(2) \end{bmatrix} &= \lambda_2 \left(\begin{bmatrix} q_1(1) \\ q_1(2) \end{bmatrix} - \begin{bmatrix} q_2(1) \\ q_2(2) \end{bmatrix} \right) + \begin{bmatrix} q_2(1) \\ q_2(2) \end{bmatrix} \iff \\
\begin{bmatrix} \lambda_1(p_1(1) - p_2(1)) + p_2(1) \\ \lambda_1(p_1(2) - p_2(2)) + p_2(2) \end{bmatrix} &= \begin{bmatrix} \lambda_2(q_1(1) - q_2(1)) + q_2(1) \\ \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{bmatrix} \quad (1)
\end{aligned}$$

If $p_1(1) - p_2(1) = 0$ then $p_1(2) - p_2(1) \neq 0$ because $\mathbf{p}_1 \neq \mathbf{p}_2$ and $q_1(1) - q_2(1) \neq 0$ because $(q_2(1) - q_1(1))(p_2(2) - q_1(2)) - (q_2(2) - q_1(2))(p_2(1) - q_1(1)) \neq 0$

$$\begin{aligned}
(1) &\iff \begin{bmatrix} p_2(1) \\ \lambda_1(p_1(2) - p_2(2)) + p_2(2) \end{bmatrix} = \begin{bmatrix} \lambda_2(q_1(1) - q_2(1)) + q_2(1) \\ \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{bmatrix} \iff \\
&\begin{cases} \lambda_2 = \frac{p_2(1) - q_2(1)}{q_1(1) - q_2(1)} \\ \lambda_1(p_1(2) - p_2(2)) + p_2(2) = \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{cases} \iff \\
&\begin{cases} \lambda_2 = \frac{p_2(1) - q_2(1)}{q_1(1) - q_2(1)} \\ \lambda_1(p_1(2) - p_2(2)) + p_2(2) - q_2(2) = \frac{p_2(1) - q_2(1)}{q_1(1) - q_2(1)}(q_1(2) - q_2(2)) \end{cases} \iff \\
&\begin{cases} \lambda_2 = \frac{p_2(1) - q_2(1)}{q_1(1) - q_2(1)} \\ \lambda_1 = \frac{p_2(1) - q_2(1)(q_1(2) - q_2(2)) - (p_2(2) - q_2(2))(q_1(1) - q_2(1))}{(p_1(2) - p_2(2))(q_1(1) - q_2(1))} \end{cases}
\end{aligned}$$

Else if $p_1(1) - p_2(1) \neq 0$

$$\begin{aligned}
(1) &\iff \begin{cases} \lambda_1(p_1(1) - p_2(1)) + p_2(1) = \lambda_2(q_1(1) - q_2(1)) + q_2(1) \\ \lambda_1(p_1(2) - p_2(2)) + p_2(2) = \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{cases} \iff \\
&\begin{cases} \lambda_1 = \frac{\lambda_2(q_1(1) - q_2(1)) + q_2(1) - p_2(1)}{p_1(1) - p_2(1)} \\ \frac{\lambda_2(q_1(1) - q_2(1)) + q_2(1) - p_2(1)}{p_1(1) - p_2(1)}(p_1(2) - p_2(2)) + p_2(2) = \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{cases} \iff \\
&\begin{cases} \lambda_1 = \frac{\lambda_2(q_1(1) - q_2(1)) + q_2(1) - p_2(1)}{p_1(1) - p_2(1)} \\ \frac{\lambda_2(q_1(1) - q_2(1))}{p_1(1) - p_2(1)}(p_1(2) - p_2(2)) + \frac{q_2(1) - p_2(1)}{p_1(1) - p_2(1)}(p_1(2) - p_2(2)) + p_2(2) = \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{cases} \\
&\begin{cases} \lambda_1 = \frac{\lambda_2(q_1(1) - q_2(1)) + q_2(1) - p_2(1)}{p_1(1) - p_2(1)} \\ \frac{q_2(1) - p_2(1)}{p_1(1) - p_2(1)}(p_1(2) - p_2(2)) + p_2(2) = \lambda_2[(q_1(2) - q_2(2)) - \frac{(q_1(1) - q_2(1))}{p_1(1) - p_2(1)}(p_1(2) - p_2(2))] + q_2(2) \end{cases} \\
&\begin{cases} \lambda_1 = \frac{\lambda_2(q_1(1) - q_2(1)) + q_2(1) - p_2(1)}{p_1(1) - p_2(1)} \\ \lambda_2 = \frac{(q_2(1) - p_2(1))(p_1(2) - p_2(2)) + (p_1(1) - p_2(1))(p_2(2) - q_2(2))}{(p_1(1) - p_2(1))(q_1(2) - q_2(2)) - (q_1(1) - q_2(1))(p_1(2) - p_2(2))} \end{cases} \iff \\
&\begin{cases} \lambda_1 = \frac{(q_2(1) - p_2(1))(p_1(2) - p_2(2)) + (p_1(1) - p_2(1))(p_2(2) - q_2(2))}{(p_1(1) - p_2(1))(q_1(2) - q_2(2)) - (q_1(1) - q_2(1))(p_1(2) - p_2(2))} \frac{(q_1(1) - q_2(1)) + q_2(1) - p_2(1)}{p_1(1) - p_2(1)} \\ \lambda_2 = \frac{(q_2(1) - p_2(1))(p_1(2) - p_2(2)) + (p_1(1) - p_2(1))(p_2(2) - q_2(2))}{(p_1(1) - p_2(1))(q_1(2) - q_2(2)) - (q_1(1) - q_2(1))(p_1(2) - p_2(2))} \end{cases}
\end{aligned}$$

In both these cases if $\lambda_1, \lambda_2 \in [0, 1]$, the line segments intersect and we return

TRUE, else we return FALSE.