PLH 415

Exercise Set 1

Alevrakis Dimitrios 2017030001

Problem 1 The idea behind the range query on a binary search tree, is doing two searches on the lower and upper limits and then returning all the nodes of the right subtree of lower limit and the nodes of the left subtree of the upper limit.

Therefore assuming the range query in the space $[a, b]$.

We need to do a search for "a". That means going down the left subtree until the node key $k \leq a$. If $k < a$ we go up to the parent node, else we stay at that node. This search in general has a time complexity of $O(h)$ where h the BST tree height. Then we need to return the nodes of the right subtree of found node, which has a time complexity of $O(k)$.

The algorithm for the search of upper limit node follows the same idea by, this time, going down the right subtree until the node key $k \geq a$. If $k > a$ we go up to the parent node, else we stay at that node. This search has also a time complexity of $O(h)$. Then, again, we need to return the nodes of the left subtree of found node, which has a time complexity of $O(k)$.

Therefore the algorithm complexity is $O(n) + O(k)O(n) + O(k) = O(n + k)$

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:
Let $\mathbf{p_1} = \begin{bmatrix} p_1(1) \\ p_1(2) \end{bmatrix}, \mathbf{p_2} = \begin{bmatrix} p_2(1) \\ p_2(2) \end{bmatrix}, \mathbf{q_1} = \begin{bmatrix} q_1(1) \\ q_1(2) \end{bmatrix}, \mathbf{q_2} = \begin{bmatrix} q_2(1) \\ q_2(2) \end{bmatrix}$ 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:

$$|(\mathbf{q_2} - \mathbf{q_1}) \times (\mathbf{q_2} - \mathbf{q_1})| = (q_2(1) - q_1(1))(p_2(2) - q_1(2)) - (q_2(2) - q_1(2))(p_2(1) - q_1(1))$$

- If $(q_2(1) - q_1(1))(p_2(2) - q_1(2)) - (q_2(2) - q_1(2))(p_2(1) - q_1(1)) = 0$ 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 $\lambda \in$

$[0, 1]$:

$$\mathbf{p_1} = \lambda_1(\mathbf{q_1} - \mathbf{q_2}) + \mathbf{q_2} \iff \begin{cases} p_1(1) = \lambda_{1,x}(q_1(1) - q_2(1)) + q_2(1) \\ p_1(2) = \lambda_{1,y}(q_1(2) - q_2(2)) + q_2(2) \end{cases}$$

$$\mathbf{p_2} = \lambda_2(\mathbf{q_1} - \mathbf{q_2}) + \mathbf{q_2} \iff \begin{cases} p_2(1) = \lambda_{2,x}(q_1(1) - q_2(1)) + q_2(1) \\ p_2(2) = \lambda_{2,y}(q_1(2) - q_2(2)) + q_2(2) \end{cases}$$

$$\mathbf{q_1} = \lambda_3(\mathbf{p_1} - \mathbf{p_2}) + \mathbf{p_2} \iff \begin{cases} q_1(1) = \lambda_{3,x}(p_1(1) - p_2(1)) + p_2(1) \\ q_1(2) = \lambda_{3,y}(p_1(2) - p_2(2)) + p_2(2) \end{cases}$$

$$\mathbf{q_2} = \lambda_4(\mathbf{p_1} - \mathbf{p_2}) + \mathbf{p_2} \iff \begin{cases} q_2(1) = \lambda_{4,x}(p_1(1) - p_2(1)) + p_2(1) \\ q_2(2) = \lambda_{4,y}(p_1(2) - p_2(2)) + p_2(2) \end{cases}$$

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

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

$$\lambda_1(\mathbf{p_1} - \mathbf{p_2}) + \mathbf{p_2} = \lambda_2(\mathbf{q_1} - \mathbf{q_2}) + \mathbf{q_2} \iff$$

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

$$\begin{bmatrix} \lambda_1(p_1(1) - p_2(1)) + p_2(1) \\ \lambda_1(p_1(2) - p_2(2)) + p_2(2) \end{bmatrix} = \begin{bmatrix} \lambda_2(q_1(1) - q_2(1)) + q_2(1) \\ \lambda_2(q_1(2) - q_2(2)) + q_2(2) \end{bmatrix} \tag{1}$$

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 $q_(1) - q_2(1) \neq 0$ because $(q_2(1) - q_1(1))(p_2(2) - q_1(2)) - (q_2(2) - q_1(2))(p_2(1) - q_1(1)) \neq 0$

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

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

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

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

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

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

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

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

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