CS 240: Data Structures and Data Management **Winter 2023**

Tutorial 9: March 20

1. [E] Consider the following points being stored in a 2D range tree: $(2, 12)$, $(17, 77)$, $(23, 92)$, $(40, 47)$, $(55, 91)$, $(67, 27)$, $(89, 79)$, $(99, 53)$, $(10, 23)$, $(35, 7)$, $(61, 40)$, $(95, 56)$, $(22, 42)$, $(88, 15)$, $(42, 2)$.

a) Draw the primary tree for this range tree.

b) Draw the corresponding associate trees for the points $(88, 15)$, $(61, 40)$ and $(67, 27)$.

c) Perform a range-search with the query rectangle $[35, 90] \times [5, 30]$, indicating all boundary nodes and topmost inside nodes.

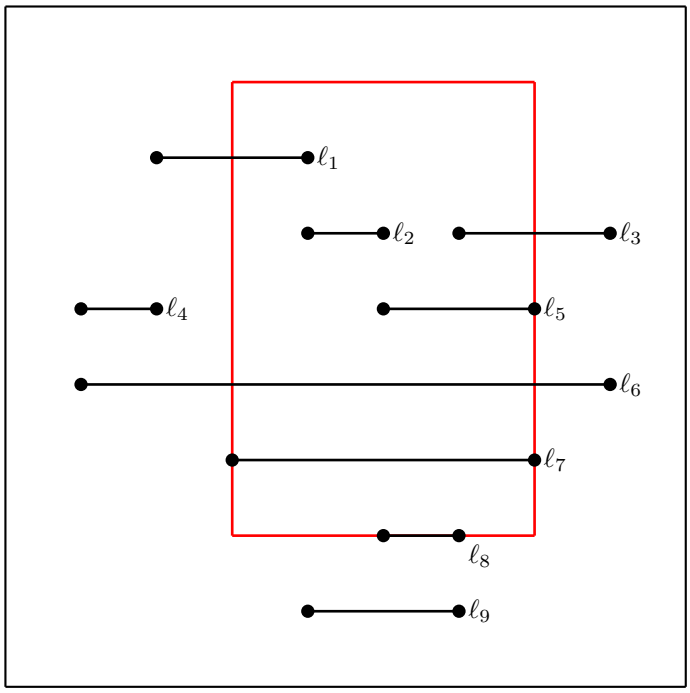
d) **Programming Challenge:** In the file `range_tree.cpp`, implement the function `topmost_inside_nodes`, which returns all the topmost inside nodes in a 1D range tree for a given range query.

2. [H] Suppose you have a set of n horizontal line segments in a plane, where line segment ℓ_i has coordinates (x_i, y_i) and (x'_i, y_i). Assume that all coordinates are integers.

For each of the range-search queries below, design a data structure and provide an algorithm to answer the queries in $O(\log^3 n + s)$ time, where s is the number of lines reported. Each range-search query is a rectangle of the form $[a, b] \times [c, d]$.

a) The algorithm reports all line segments that are entirely contained inside the query rectangle. For the example below, the algorithm would return ℓ_2, ℓ_5, ℓ_7 and ℓ_8.

b) The algorithm reports all line segments that intersect the query rectangle. For the example below, the algorithm reports all line segments **except** ℓ_4 and ℓ_9.



3. [E] For Karp-Rabin pattern matching, consider the following hash function for strings over the alphabet $\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$:

$$\begin{aligned} h(P) =& (\#\text{ of occurrences of } \texttt{A}) + 2 \times (\#\text{ of occurrences of } \texttt{C}) \\ &+ 3 \times (\#\text{ of occurrences of } \texttt{G}) + 4 \times (\#\text{ of occurrences of } \texttt{T}) \end{aligned}$$

Given the pattern $P = \texttt{TAGCAT}$ and sequence $T = \texttt{TGCCGATGTAGCTAGCAT}$, use the table below to show all the character comparisons performed during Karp-Rabin pattern matching. Start a new pattern shift (in which character comparison occurs) in a new row. You may not need all the available space.

T	G	C	C	G	A	T	G	T	A	G	C	T	A	G	C	A	T

Table 1: Table for Karp-Rabin problem.