Computational Geometry.
Assignment 3: Trapezoidal Maps.

Assignment report:

Classes:
Point: class for point
Segment : class for segment.
Trapezoid: class for trapezoid.
TrapTree : data structure for the map.
TrapezoidalMaps : main class

Interface: Node: implemented by Point, Segment and Trapezoid.

The trapezoid map is stored in a tree structure with Nodes that can be of the type of a Point Segment or Trapezoid.
Every time a segment is added the tree is to be parsed to check where the segment lies.
Three cases are countered.

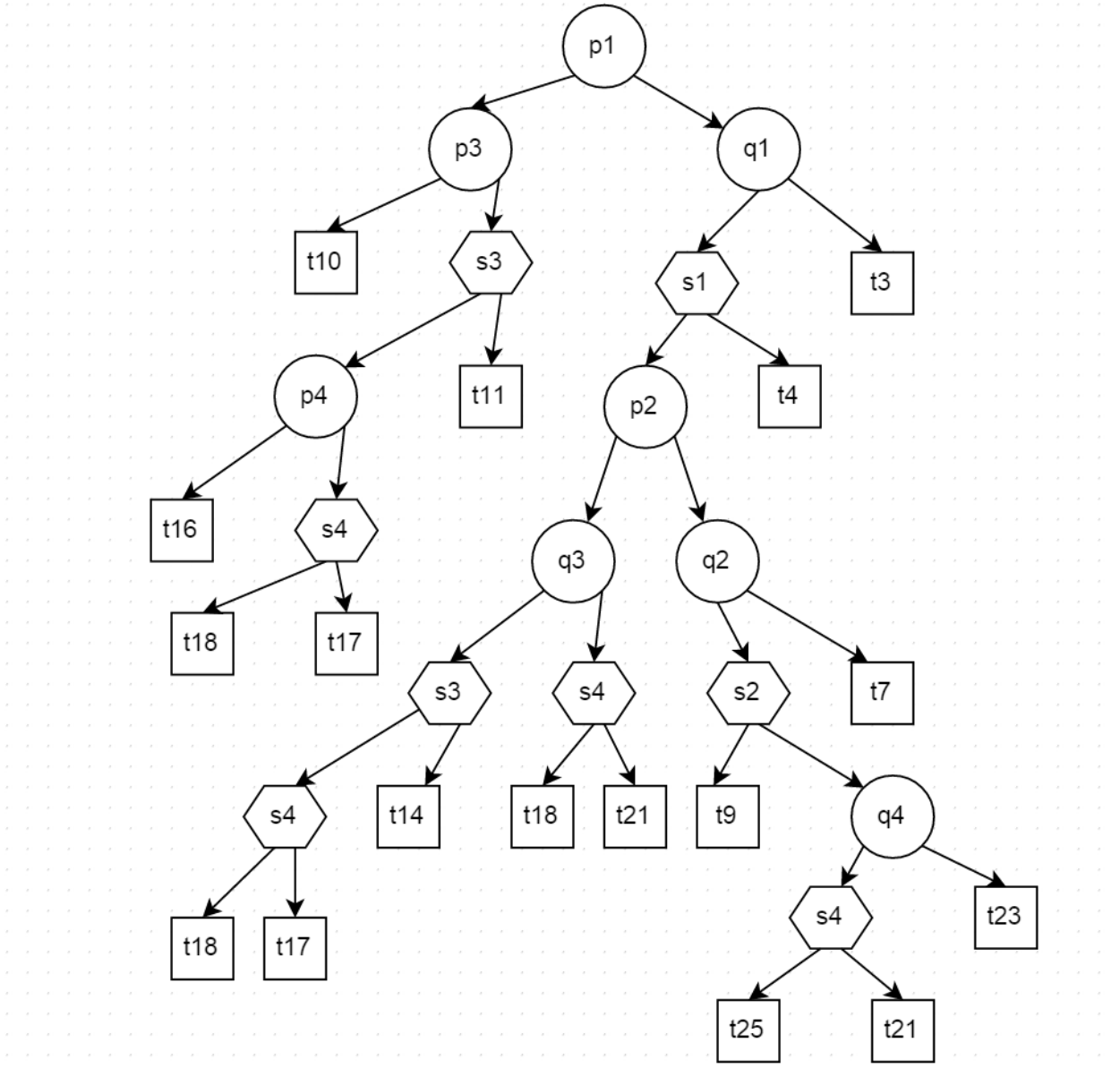
1. When the segment lies completely inside one Trapezoid. This trapezoid is broken into four trapezoids. Left, right, up and down. This case has sub cases as 1a and 1b.
2. When one point of the segment lies inside one trapezoid and the rest in one or more trapezoids.
3. When the starting and ending point of the added segment lie outside a trapezoid on either sides, cutting the trapezoid into two.

The parsing of the tree is done recursively passing control to the child nodes (left or right) depending on which side the segment's starting point lies on.

The runtime complexity of this incremental algorithm is O(n log n).
Where n is the number of segments added.
O(log n) time is taken for the tree traversal to find in which trapezoids the segment lies. This is done n times for the n segments.
This leading to O(n log n).

The first node added is the bounding trapezoid that we begin with.
Segments are added on by one.
The trapezoids are always leaf nodes and never have left or right childs.
The output is an adjacency matrix in a txt file.

The Tree for the input “ss4017.txt”



Trapezoid Map: