Assignment 4 - Trapezoidal Map, Arrangements and Duality

Duy Pham - 0980384
Mazen Aly - 0978251
Pattarawat Chormai - 0978675

December 8, 2015

2

a

Search path to q at D_j becomes longer if q is in a trapezoid that was just created by the latest insertion. We also know that at most 4 line segments define that trapezoids. Thus, the probability that the search part becomes longer is :

$$Pr[\text{Search Path to } q \text{ becomes longer at step } i] = 4/i$$

Hence, the expected increment of the length of the search part at step k, comparing to step j, where $j < k$, is:

$$\begin{aligned}
\text{Expected Length} &\leq \sum_{i=j}^{k}(4/i) \\
&= 4(\sum_{i=1}^{k}(1/i) - \sum_{i=1}^{j}(1/i)) \\
&\leq 4(1 + \ln k - 1 - \ln j) \\
&= O(\log(k/j))
\end{aligned}$$

Therefore, the expected time locating q at D_k is $O(\log(k/j))$.

b

We observe that vertical decomposition lines of e_i intersects properly with P if the edges that they intersect with have not been added into T_i yet. Hence, the number of proper intersections at T_i is :

$$\text{Number of Proper Intersections of } T_i = Pr[T_i \text{ has proper intersections}] = (n-j)/(n-1)$$

Thus, the expected number of proper intersections, E, between T_j and P is:

$$\begin{aligned}
\mathrm{E} &= \sum_{i=1}^{n}(n-i)/(n-1) \\
&= \frac{1}{n-1}\frac{(n-1)n}{2} \\
&= \frac{n}{2} \\
&= O(n)
\end{aligned}$$

c

We analyze the expected running time of the algorithm by dividing it into subprocesses.

For Line 4 - 5, the insertion takes $O(\log(i/2^{(h-1)^2}))$ as being proven in (a). Hence, in total expected running time of the process takes, $R_{4\to5}$, is :

$$\begin{aligned}
R_{4\to5} &= \sum_{i=2^{(h-1)^2}}^{2^{h^2}} O(i/2^{(h-1)^2}) \\
&\le \sum_{i=2^{(h-1)^2}}^{2^{h^2}} O(\log(i/2^{(h-1)^2})) \\
&= \frac{1}{2^{(h-1)^2}}(\sum_{i=1}^{2^{h^2}} O(i) - \sum_{i=1}^{2^{(h-1)^2}} O(i)) \\
&= \frac{1}{2^{(h-1)^2}}(\frac{2^{h^2}(2^{h^2}+1)}{2} - \frac{2^{(h-1)^2}(2^{(h-1)^2}-1)}{2}) \\
&= \frac{O(2^{2h^2})}{2^{(h-1)^2}} \\
&= O(2^{h^2})
\end{aligned}$$

For Line 6, everytime a line segment is split by a vertical line, we have to make a step in T to find the trapezoids containing the endpoints. Thus, the running time of finding the trapezoid containing each vertex of P depends on the number of proper intersections between P and T. According to part (b), R_6 is:

$$R_6 = O(n)$$

Hence $R_{3\to6}$ is:

$$
\begin{aligned}
R_{3\to 6} &= \sum_{h=1}^{\sqrt{\log n}} (O(2^{h^2}) + O(n)) \\
&= O(n\sqrt{\log n})
\end{aligned}
$$

For Line 8, because we start from D_n, hence, R_8 is $O(n)$ and we do Line 7 only 2 times, $n+1$ to n. Thus, $R_{7\to 8}$ is $O(1)$.

Therefore, the expected running time of the algorithm is $O(n\sqrt{\log n})$.

3

a

Let s_{i^*} be the region in the dual plane corresponding to a line segment in S and l^* is the point of l in the dual plane. Thus, the problem can be formulated as finding the number of $s_{i^*} \in S^*$ that p^* belongs to.

b

Now the input turns out to be a set of n pairs of intersected lines and a query point p. We know that arrangement of the segments $A(S^*)$ has complexity $O(n^2)$, $O(n^2)$ vertices and edges. Thus, we can use a DAG D which is used in vertical decomposition as a query data structure. Every trapezoid is stored in a leave of D, and stores the information about how many regions (corresponding to the original line segments) it belongs to.

Because we build D from $A(S^*)$, the expected storage complexity of D becomes $O(n^2)$. Moreover, the depth of D is $O(\log n^2)$, thus, the expected query time for searching a point is still $O(\log n)$.

c