More on
RIC Point Location

Dan Halperin, Tel Aviv University

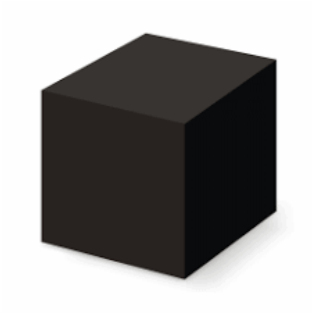
Deterministic guarantees

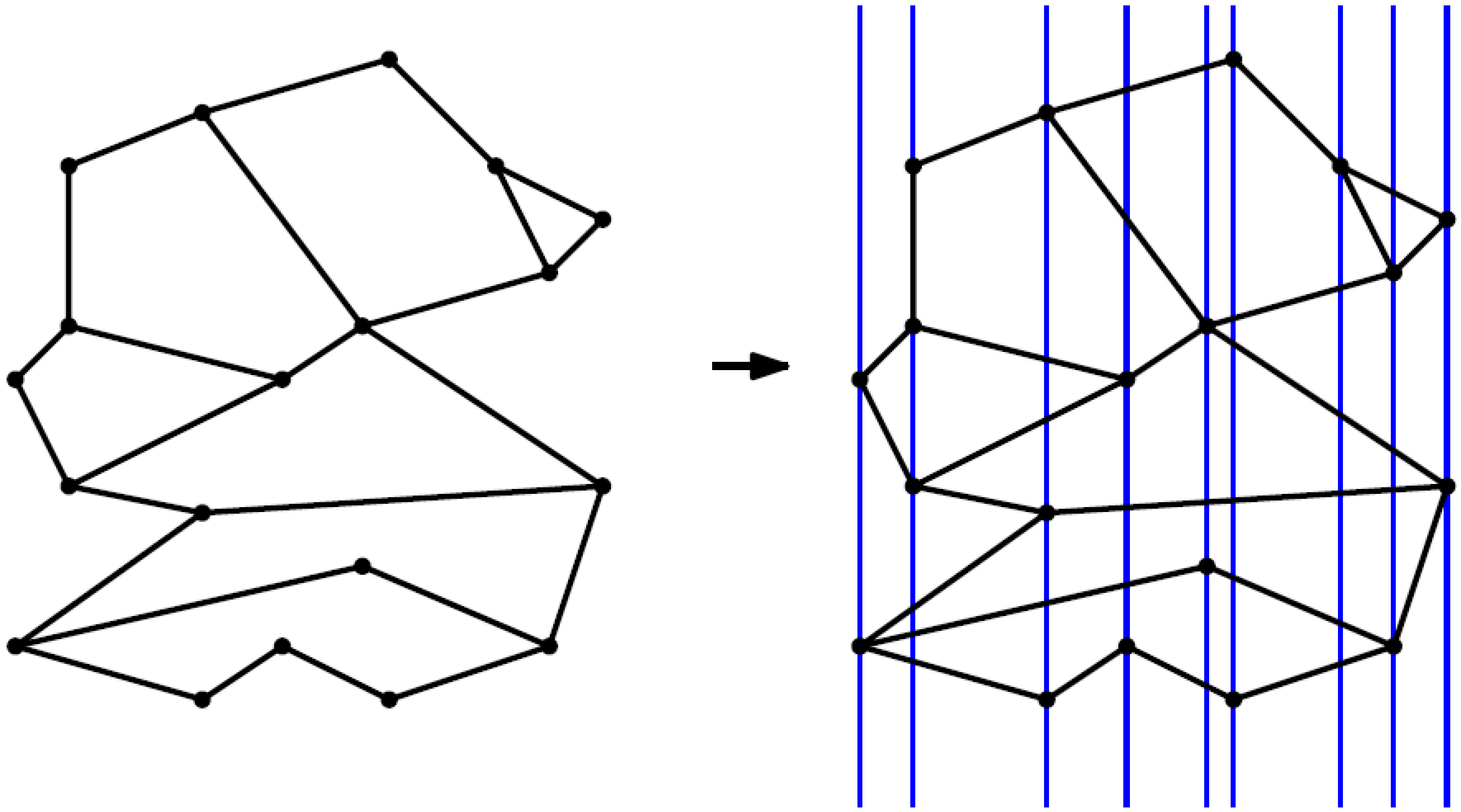
for storage and query time

Tail estimate

Lemma 6.6 *Let S be a set of n non-crossing line segments, let q be a query point, and let λ be a parameter with $\lambda > 0$. Then the probability that the search path for q in the search structure computed by Algorithm* TRAPEZOIDALMAP *has more than $3\lambda \ln(n+1)$ nodes is at most $1/(n+1)^{\lambda \ln 1.25 - 1}$.*

[CGAA]





[L9vK]

Maximum query time/length

Lemma 6.7 *Let S be a set of n non-crossing line segments, and let λ be a parameter with $\lambda > 0$. Then the probability that the maximum length of a search path in the structure for S computed by Algorithm* TRAPEZOIDALMAP *is more than $3\lambda \ln(n+1)$ is at most $2/(n+1)^{\lambda \ln 1.25 - 3}$.*

[CGAA]

- For example, for $\lambda = 20$ and $n > 4$ this probability is smaller than $\frac{1}{4}$
- Namely, with probability at least $\frac{3}{4}$ we get a structure with good query length, bounded by $c_1 \log n$

Maximum storage space

- Following similar analysis*, we can show that with probability at least $\frac{3}{4}$ we obtain a structure requiring at most $c_2 n$ storage space

* Michael Hemmer, Michal Kleinbort, Dan Halperin:
Optimal randomized incremental construction for guaranteed logarithmic planar point location. Comput. Geom. 58: 110-123 (2016)

Question

- Can we use these observations to derive an efficient algorithm to produce a point location structure with guaranteed deterministic worst-case bound on query time and storage?

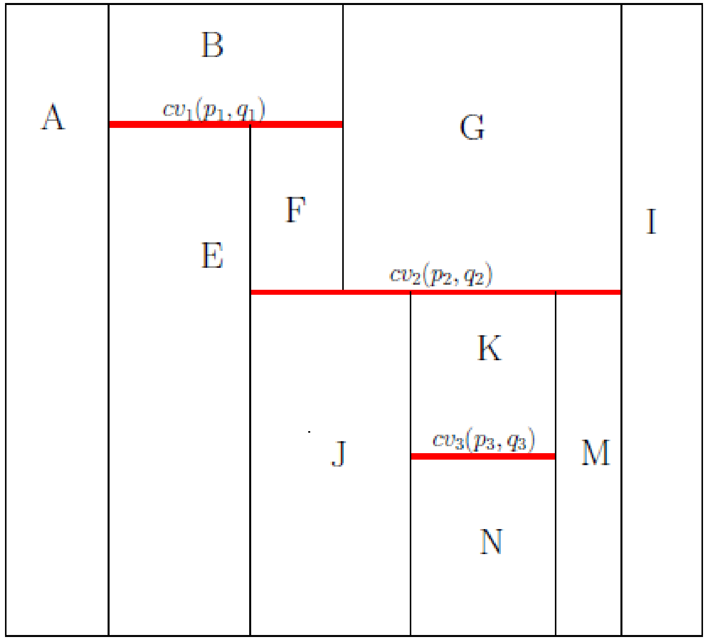
breakout

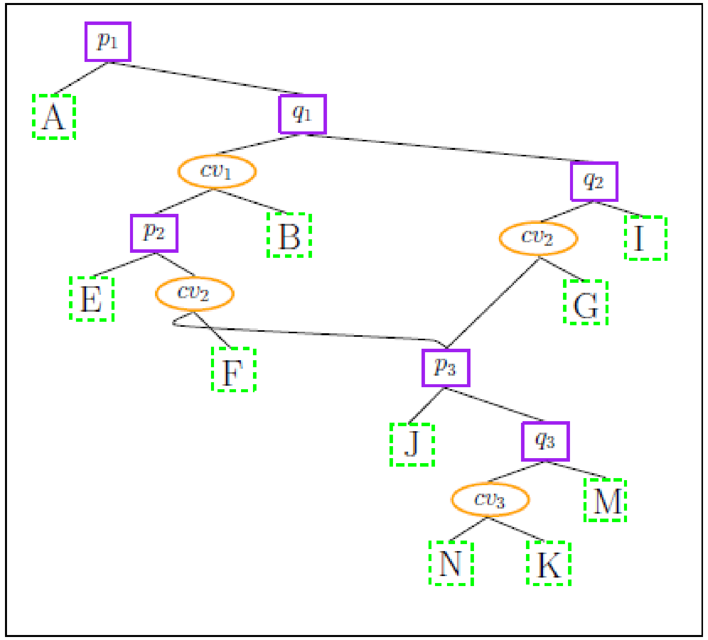
Scheme

- Monitor the maximum query length so far during a run of the RIC algorithm; abort if $\geq c_1 \log n$
- Monitor the storage allocated so far during a run of the RIC algorithm; abort if $\geq c_2 n$

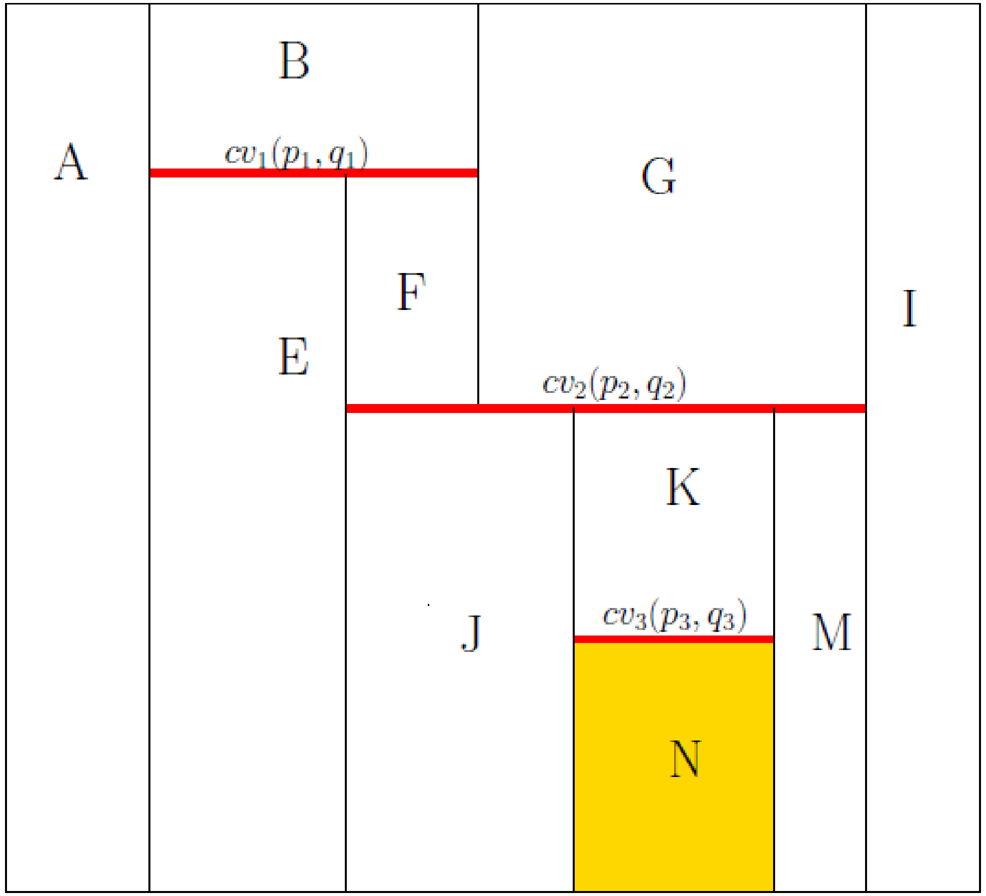
The catch

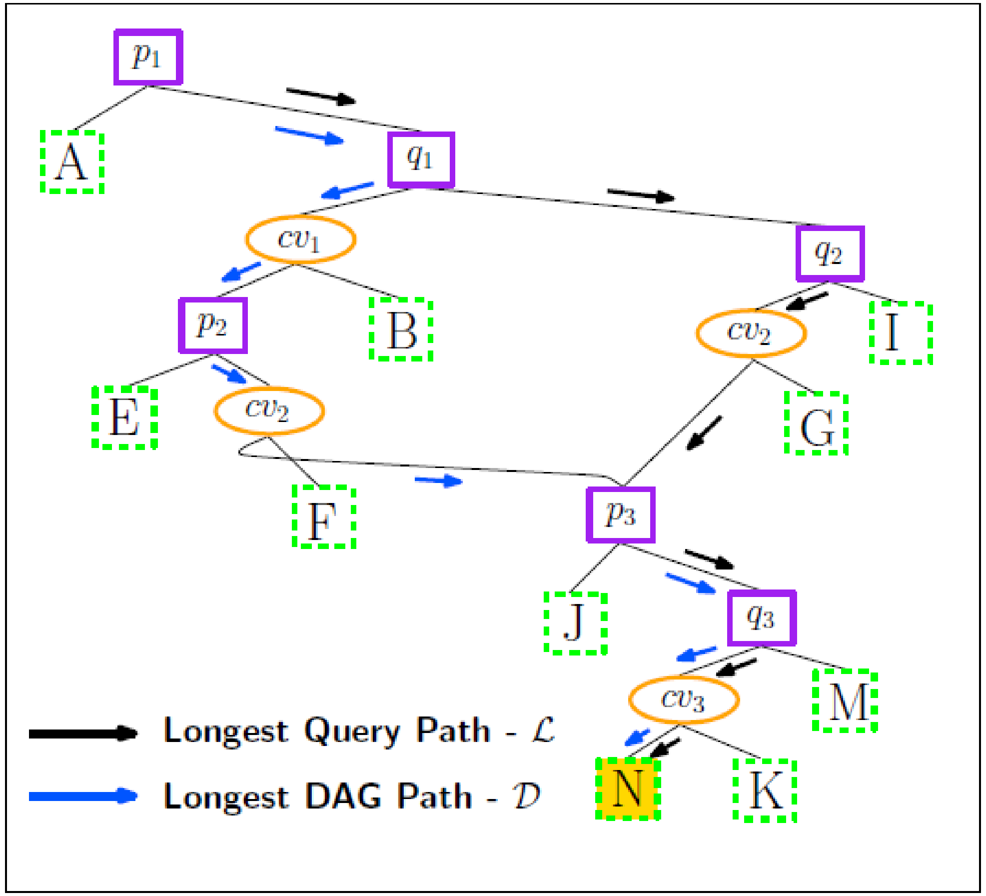
- It is easy to monitor the storage allocated so far during a run of the RIC algorithm
- How do we monitor the maximum query length?





Longest query path vs. longest path



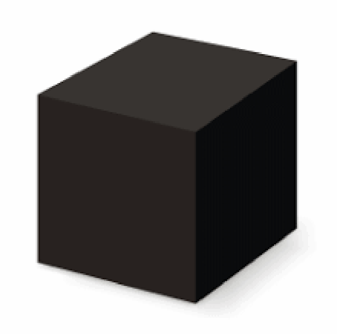


- L: the longest query path in the search structure
- D: the longest path in the search structure

- There are search structures for which
  - L is $O(\log n)$, and
  - D is $\Omega(n)$

Guaranteed logarithmic time Point Location

- The length L in a linear size DAG can be verified in $O(n \log n)$ time



- A point location data structure for a planar subdivision with n edges, which has $O(n)$ size and $O(\log n)$ query time in the worst case, can be built in expected $O(n \log n)$ time

Reference

Michael Hemmer, Michal Kleinbort, Dan Halperin:
Optimal randomized incremental construction for guaranteed logarithmic planar point location. Comput. Geom. 58: 110-123 (2016)

THE END