07-I-05

Geometric Range Search

Range Tree: Optimization

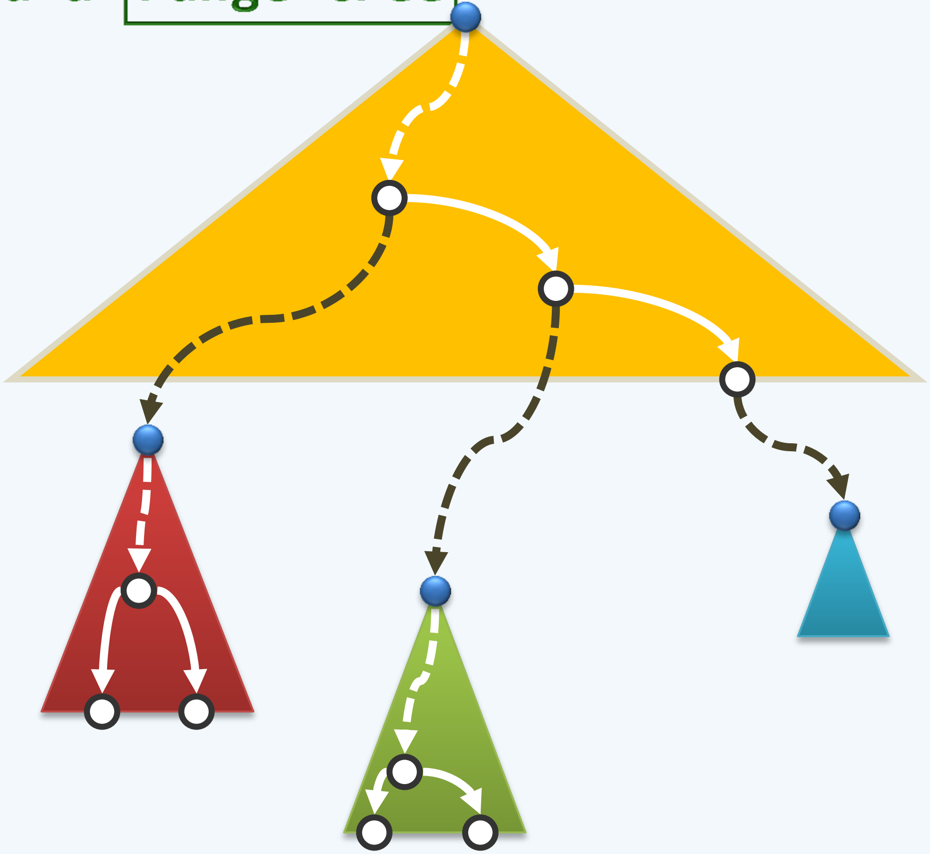
- Complexity

Junhui DENG

deng@tsinghua.edu.cn

Complexity

- An MLST with fractional cascading is called a $\boxed{\text{range tree}}$
- At the root of the main tree,
  - we need to perform a binary search
    with all the $\boxed{y}$-values
    to determine which points
    lie within this interval, and
  - it requires $\boxed{\mathcal{O}(\log n)}$ time
- For all subsequent levels, once
  we know where the $\boxed{y}$-interval falls w.r.t. to the order points here,
  we can drop down to the next level in $\boxed{\mathcal{O}(1)}$ time



Complexity

- Thus, as with fractional cascading,

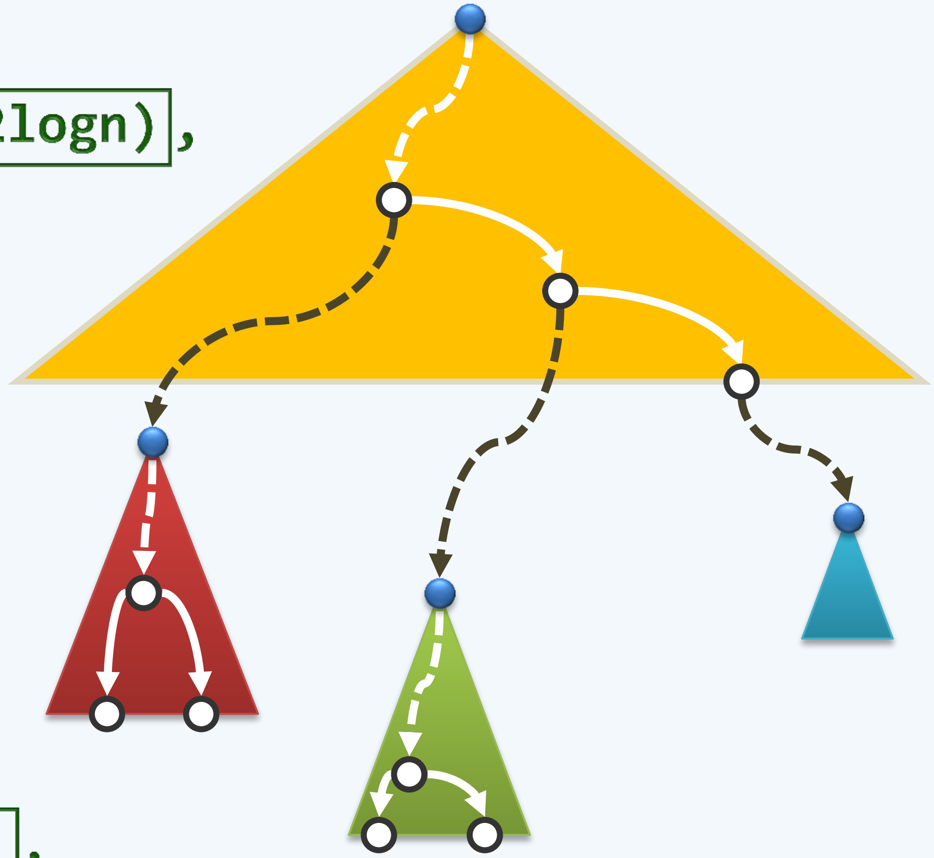
  the time for cascading searches is $\boxed{\mathcal{O}(2\log n)}$,

  rather than $\boxed{\mathcal{O}(\log^2 n)}$

- Given a set of n points in the plane,

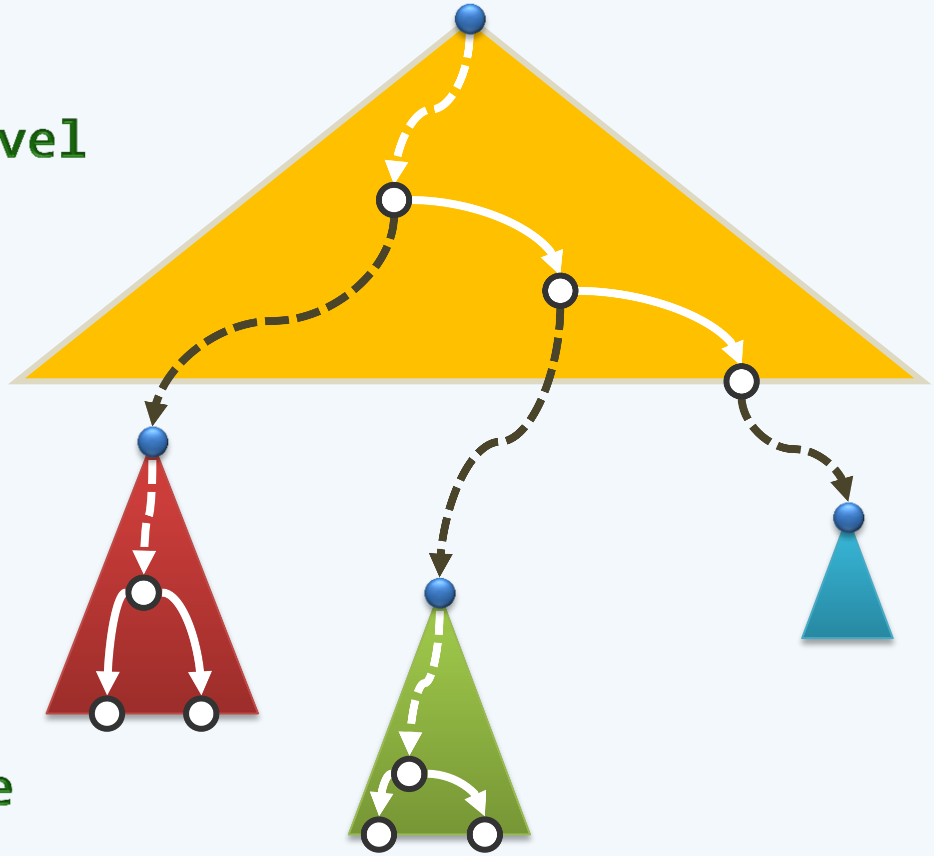
  orthogonal range queries

  - can be answered in $\boxed{\mathcal{O}(r + \log n)}$ time
  - from a data structure of size $\boxed{\mathcal{O}(n\log n)}$,
  - which can be constructed in $\boxed{\mathcal{O}(n\log n)}$ time



Beyond 2D

- Unfortunately, it turns out that
  the trick of fractional cascading
  can $\boxed{\text{only}}$ be applied to the $\boxed{\text{last}}$ level
  of the search structure,
  because all other levels need
  the $\boxed{\text{full}}$ tree search
  to compute canonical sets



- Given a set of n points in $\mathcal{E}^d$,
  an orthogonal range query
  - can be answered in $\boxed{\mathcal{O}(r + \log^{(d-1)} n)}$ time
  - from a data structure of size $\boxed{\mathcal{O}(n * \log^{(d-1)} n)}$,
  - which can be constructed in $\boxed{\mathcal{O}(n * \log^{(d-1)} n)}$ time