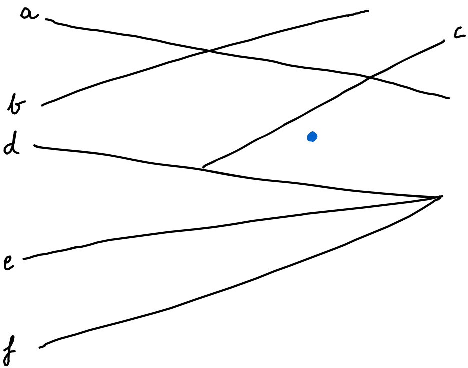
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Gerth S. Brodal Casper M. Rysgaard Jens Kristian R Schou Rolf Svenning

SOLIDUM PETIT IN PROFUNDIS · UNIVERSITAS ARHUSIENSIS ·

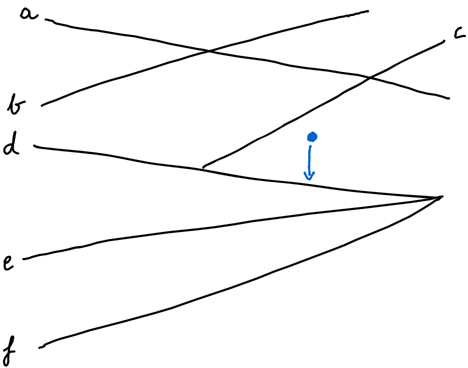
Aarhus University
Denmark

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



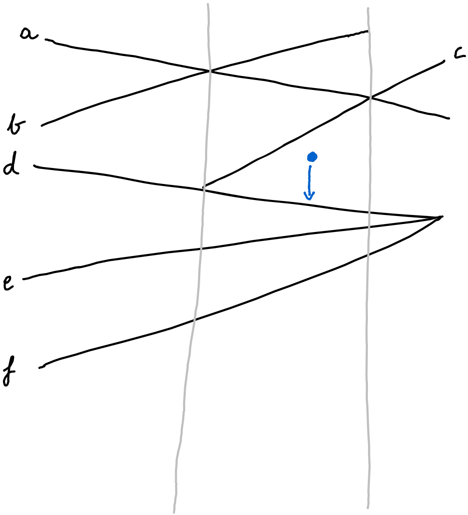
Sarnak, Tarjan 85

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



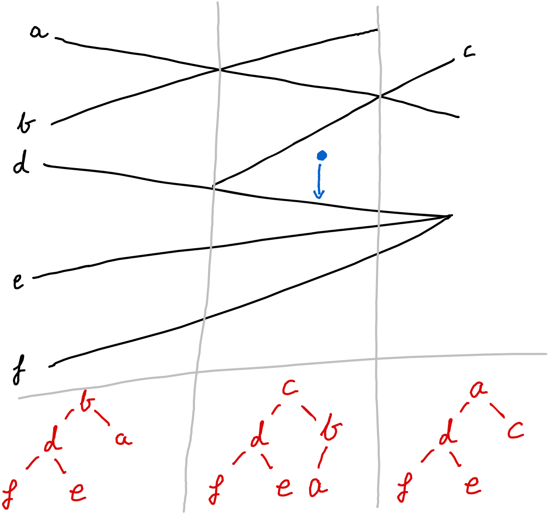
Sarnak, Tarjan 85

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



Sarnak, Tarjan 85

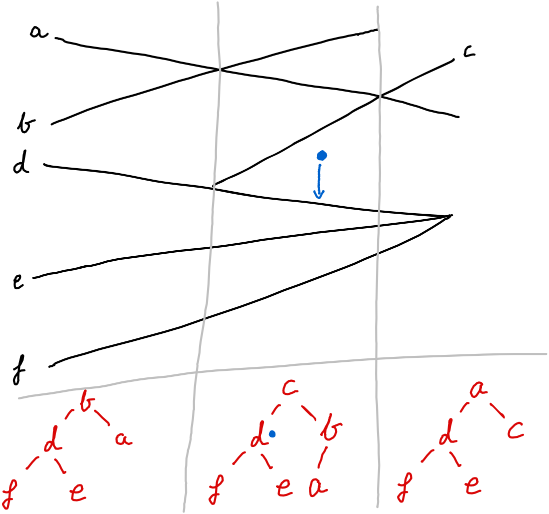
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



Sarnak, Tarjan 85

- Persistent datastructure
  - Time x axis
  - Value "aboveness"

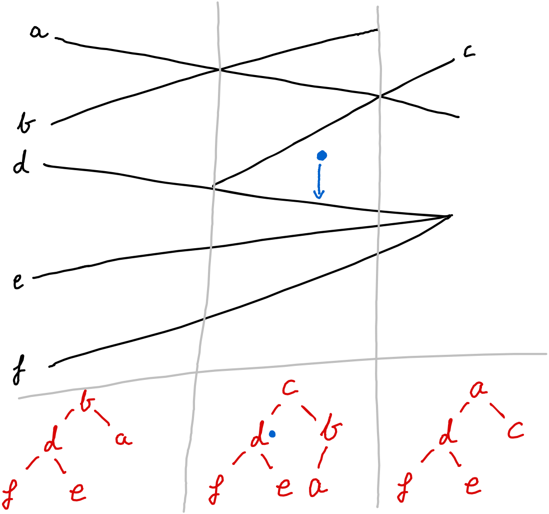
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



Sarnak, Tarjan 85

- Persistent datastructure
  - Time x axis
  - Value "aboveness"

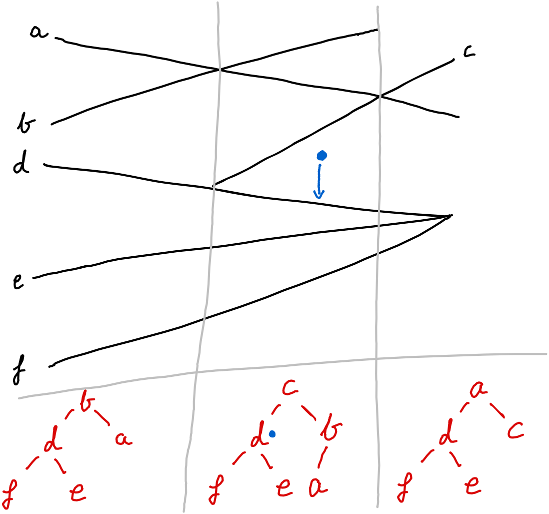
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



Sarnak, Tarjan 85

- Persistent datastructure
  - Time x axis
  - Value "aboveness"
- Partial Persistence
  - Query past versions
    - ◻ Stored as a graph

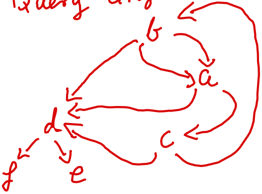
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location



Sarnak, Tarjan 85

- Persistent datastructure
  - Time x axis
  - Value "aboveness"
- Partial Persistence
  - Query past versions
    - Stored as a graph

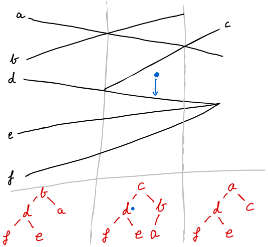
Query Graph

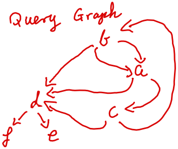


Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Sarnak, Tarjan 85

- Persistent datastructure
  - Time x axis
  - Value "aboveness"
- Partial Persistence
  - Query past versions
    - ▫ Stored as a graph
- Offline Partial Persist
  - All updates before queries





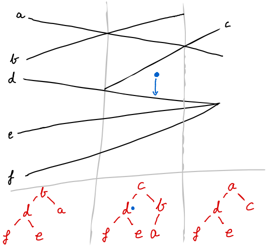
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

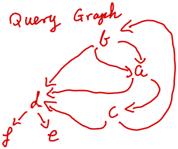
Our contribution

	Construction	Space	Query	Model
ST85	n log n	n	log n	Imperative
ST85	n log n	n log n	log n	Functional
New	n log n	n	log n	Functional

Sarnak, Tarjan 85

- Persistent datastructure
  - Time x axis
  - Value "aboveness"
- Partial Persistence
  - Query past versions
    - □ Stored as a graph
- Offline Partial Persist
  - All updates before queries

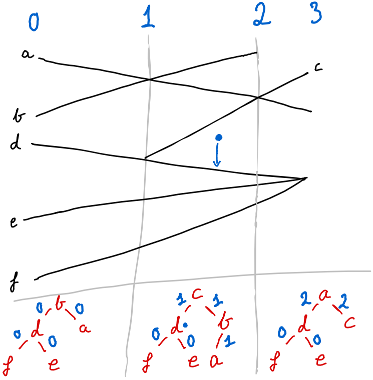


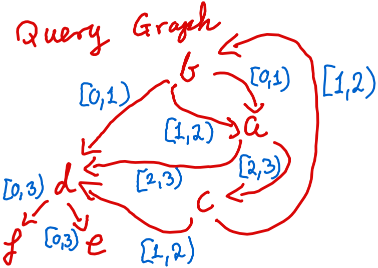


Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Imperative OPP algorithm

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Query graph with "lifespan" $[t_1, t_2)$





Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

- Functional data structures
  - Immutable
  - Inheritantly persistent
  - Change by addition
  - No cycles in graphs

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

- Functional data structures
  - Immutable
  - Inheritantly persistent
  - Change by addition
  - No cycles in graphs
- No side effects
- Excellent for recursive data structures
- Reduced model checking complexity

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

- Functional data structures
  - Immutable
  - Inheritantly persistent
  - Change by addition
  - No cycles in graphs

Functional OPP algorithm ?

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Query graph with "lifespan" $[t_1, t_2)$

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

- Functional data structures
  - Immutable
  - Inheritantly persistent
  - Change by addition
  - No cycles in graphs

Functional OPP algorithm ?

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Query graph with "lifespan" $[t_1, t_2)$

Functional challenge

1 no cycles
  - Solution: Query DAG

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

- Functional data structures
  - Immutable
  - Inheritantly persistent
  - Change by addition
  - No cycles in graphs

Functional OPP algorithm ?

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Query graph with "lifespan" $[t_1, t_2)$

Functional challenge

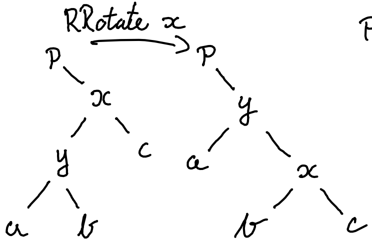
1 no cycles
  - Solution: Query DAG

2 Functional DAG insertion is $\Omega(n)$
  - Solution: Topological Sort

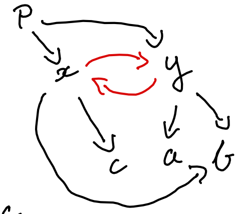
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Breaking Cycles

Red-Black tree



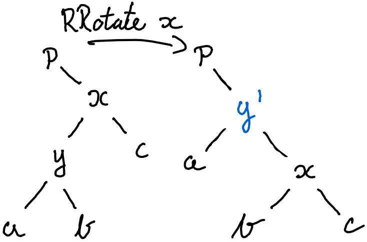
Query Graph



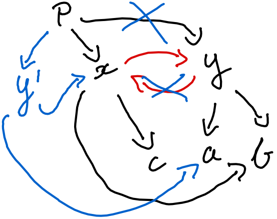
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Breaking Cycles (new)

Red-Black tree



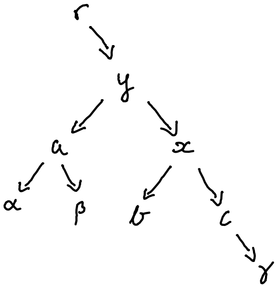
Query DAG



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

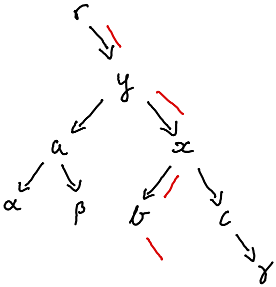
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

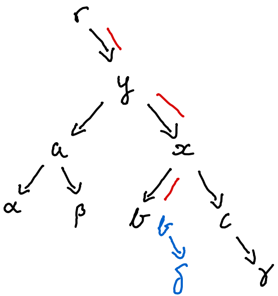
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

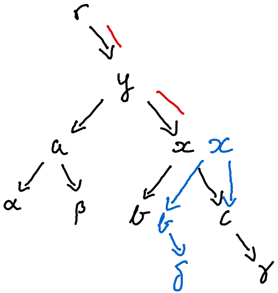
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

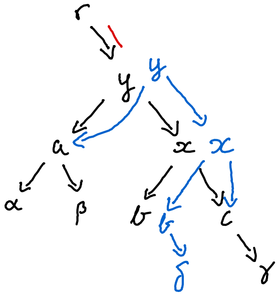
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

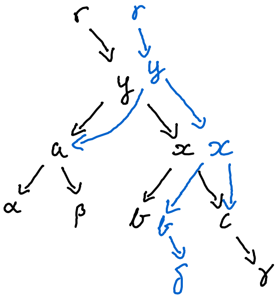
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion via path copying

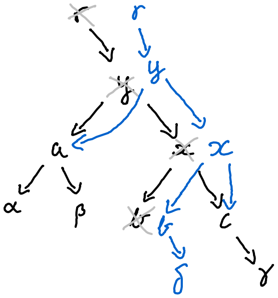
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

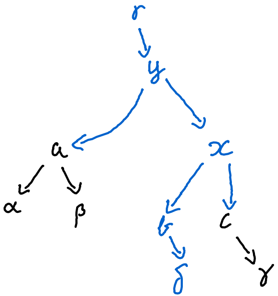
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

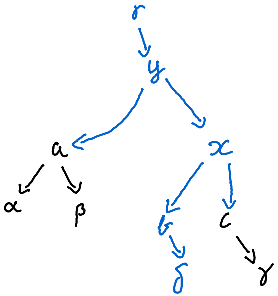
insert $\delta \in [b, x]$



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional BST insertion
via path copying

insert $\delta \in [b, x]$

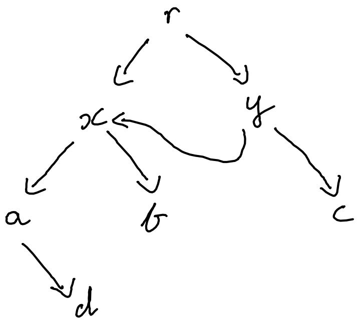


$O(\log n)$ time
$O(1)$ space

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

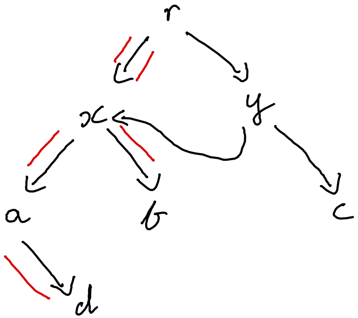
insert edge b -> d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

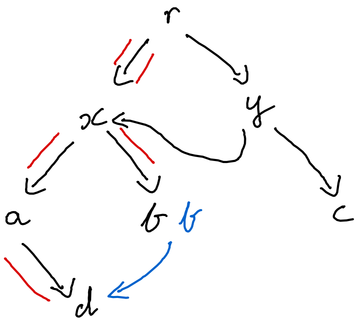
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

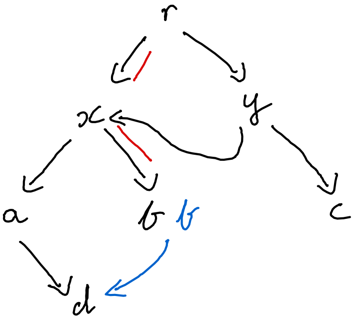
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

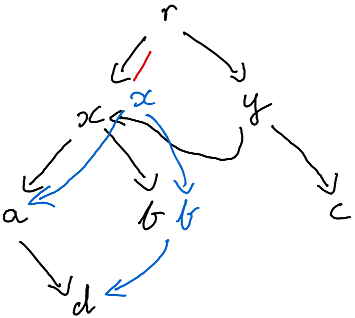
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

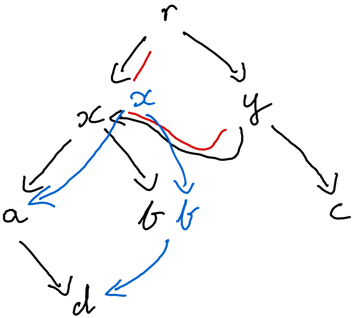
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

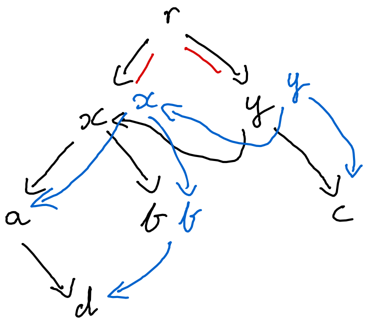
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

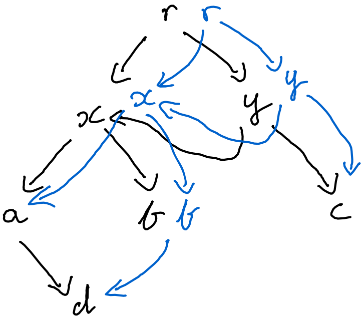
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

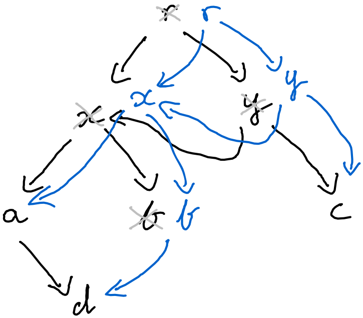
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

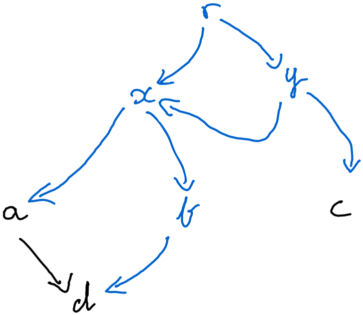
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion
via path copying

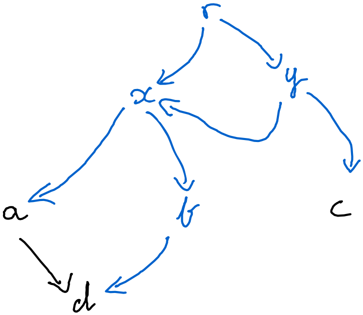
insert edge b->d



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG insertion via path copying

insert edge b->d

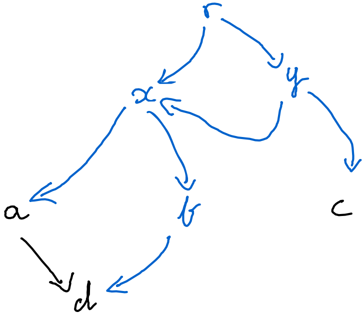


$\Omega(n)$ time (worst case)

$O(1)$ space

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG
without path copying
given all edges E
and nodes V



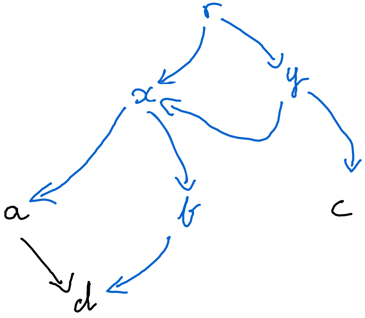
Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional Query DAG
without path copying
given all edges E
and nodes V

Algorithm

1 Topologically Sort (V, E)
[Kahn 62]

2 Construct Query DAG
bottom up



Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional OPP algorithm ?

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Query graph with "lifespan" $[t_1, t_2)$

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional OPP algorithm

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Freezer with "lifespan" $[t_1, t_2)$
- After all updates
  - Construct Functional Query DAG from edges in the Freezer

Space-Efficient Functional Offline-Partially-Persistent Trees with Applications to Planar Point Location

Functional OPP algorithm

- On edge insertion, at time t_1
  - add creation timestamp t_1
- On edge deletion, at time t_2
  - insert edge in Freezer with "lifespan" $[t_1, t_2)$
- After all updates
  - Construct Functional Query DAG from edges in the Freezer

Efficient for

- Random access arrays [Okasaki 95]
- Treaps [Aragon Seidel 89]
- Red-Black trees [Bayer 72]
- AVL-trees [Adel'son-Vel-skii, Landis 62]