| Interval | Description |
|----------|-------------|
| [1, 10)  |             |
| [5, 8)   |             |
| [6, 20)  |             |
| [15, 25) |             |
Dynamic Segment Trees  [van Kreveld, Overmars, JACM 1993]

[1, 10)    [5, 8)    [6, 20)    [15, 25)

Stabbing Query

Given a set of intervals $M$ and a point $p$, find all intervals $I \in M$ with $p \in I$.  

## Dynamic Segment Trees

[van Kreveld, Overmars, JACM 1993]

| 1   | 5   | 6   | 8   | 10  | 15  | 20  | 25  |
|-----|-----|-----|-----|-----|-----|-----|-----|
| [1, 10) | [5, 8) | [6, 20) | [15, 25) |
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees  

[1, 10)  [5, 8)  [6, 20)  [15, 25)  

[1]  [5]  [6, 8)  10)  [15, 20)  25)
Dynamic Segment Trees  [van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees

[van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]

Diagram showing segment trees with intervals [1, 10), [5, 8), [6, 20), and [15, 25).

Root node labeled 10, with children labeled [1, 5), [5, 10), [10, 20), [20, 25).
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees

[van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees  [van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]

[1, 10)  [5, 8)  [6, 20)  [15, 25)

[1]  [5]  [6]  8)  10)  [15]  20)  25)

7?

“Weak Segment Tree Property”
Dynamic Segment Trees

[van Kreveld, Overmars, JACM 1993]
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]

[1, 10) [5, 8) [6, 20) [15, 25)

Balance!
Dynamic Segment Trees [van Kreveld, Overmars, JACM 1993]

van Kreveld & Overmars’ solution
- Use Red-Black Trees
- Repair annotations after rebalancing

Balance!
Zip Trees — Insertion

[ Tarjan et al. WADS 2019. ]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zip Trees — Insertion

[Tarjan et al. WADS 2019.]
Zipping Segment Trees - Insertion

2 ➔ 5
3
1

...
Zipping Segment Trees - Insertion

Challenge
Uphold
Weak Segment Tree Property
Zipping Segment Trees - Insertion

Challenge
Uphold
Weak Segment Tree Property
Zipping Segment Trees - Insertion

Challenge

Uphold
Weak Segment Tree Property
Zipping Segment Trees - Insertion

Challenge
Uphold Weak Segment Tree Property

Idea
Clear the “unzipped” path.
Zipping Segment Trees - Insertion

Challenge

Uphold
Weak Segment Tree Property

Idea

Clear the “unzipped” path.
Zipping Segment Trees - Insertion

Challenge
Uphold Weak Segment Tree Property

Idea
Clear the “unzipped” path.
Zipping Segment Trees - Insertion

Challenge
Uphold Weak Segment Tree Property

Idea
Clear the “unzipped” path.
Zipping Segment Trees - Insertion

**Challenge**
Uphold
Weak Segment Tree Property

**Idea**
Clear the “unzipped” path.
Zipping Segment Trees - Insertion

Challenge
Uphold Weak Segment Tree Property

Idea
Clear the “unzipped” path.
Zipping Segment Trees - Insertion

Correctness (Intuition)
Look at search paths along the cleared path.

Challenge
Uphold Weak Segment Tree Property

Idea
Clear the “unzipped” path.

Correctness (Intuition)
Look at search paths along the cleared path.
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
Zipping Segment Trees - Deletion
What about search paths exiting to the right here?
What about search paths exiting to the right here?
Insertion Height

Main Idea

- We want to expect a balanced tree
- Insert node with prob. $\frac{1}{2}$ as leaf, with prob. $\frac{1}{4}$ at height 1, ...
Insertion Height

**Main Idea**
- We want to expect a balanced tree
- Insert node with prob. $\frac{1}{2}$ as leaf, with prob. $\frac{1}{4}$ at height 1, \ldots

**“Random” Variant**
- Flip a coin until hitting “heads”
Insertion Height

Main Idea
- We want to expect a balanced tree
- Insert node with prob. $\frac{1}{2}$ as leaf, with prob. $\frac{1}{4}$ at height 1, ... 

“Random” Variant
- Flip a coin until hitting “heads”

“Hashing” Variant
- Hash the node’s value (or its memory address, or ...)
- Use the bits as a stream of coin flips
- Advantage: Don’t need to store the rank at the node!
Experimental Results

1. Create tree with \( n \) random intervals (\( x \) axis)
2. Insert \( k \) new random intervals
3. \( y \) axis: Time for step 2 divided by \( k \)

- Red-Black
- Weight-Balanced
- Zip (Hashing)
- Zip (Random)
Experimental Results

1. Create tree with \( n \) random intervals (\( x \) axis)
2. Insert \( k \) new random intervals
3. \( y \) axis: Time for step 2 divided by \( k \)

![Graph showing Insertion time vs Tree Size]

- Red-Black
- Weight-Balanced
- Zip (Hashing)
- Zip (Random)

\[\text{[B, Wagner. SEA 2020.]}\]

\[\text{[B, Wagner. ALENEX 2020.]}\]
Experimental Results

1. Create tree with $n$ random intervals ($x$ axis)
2. Insert $k$ new random intervals
3. $y$ axis: Time for step 2 divided by $k$

Deletion

- Red-Black
- Weight-Balanced
- Zip (Hashing)
- Zip (Random)

[B, Wagner. SEA 2020.]
Experimental Results

1. Create tree with $n$ random intervals (x axis)
2. Insert $k$ new random intervals
3. y axis: Time for step 2 divided by $k$

[Graph showing time vs. tree size for different tree types: Red-Black, Weight-Balanced, Zip (Hashing), Zip (Random)]
Conclusion
Conclusion
Conclusion

Zipping Segment Trees
**Conclusion**

- Most efficient choice for deletions and moves
Conclusion

- Most efficient choice for deletions and moves
- Next step: Tuning Zip Trees!