

# Bachelorarbeit

Andreas Windorfer

30. Juni 2020

## Zusammenfassung

# Inhaltsverzeichnis

<b>1</b>	<b>Zipper-Baum</b>	<b>4</b>
1.1	Repräsentation eines preferred path . . . . .	4

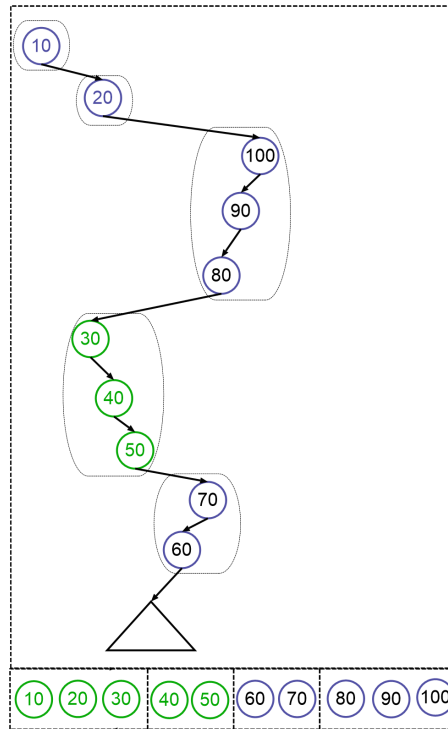
# 1 Zipper-Baum

Der Zipper-Baum basiert auf dem Tango-Baum und nutzt auch preferred paths aus einem lower-bound-tree  $P$ . Aufbau und Pflege der preferred paths in  $P$  unterscheiden sich nicht von Tango-Baum, wohl aber ihre Repräsentation im eigentlichen BST  $T$ . Der Zipper-Baum wurde in [1] vorgestellt. Er ist ebenfalls  $\log(\log(n))$ -competitive, garantiert aber auch  $O(\log(n))$  bei einer einzelnen *access* Operation.  $n$  steht wieder für die Anzahl der Knoten von  $T$ .

## 1.1 Repräsentation eines preferred path

Die Repräsentation eines preferred path  $p_1, p_2, \dots, p_p$  wird in einen top path und einem bottom tree unterteilt. Enthält der preferred path nicht mehr als  $O(\log(\log(n)))$  Knoten, wird der preferred path allein durch den top path repräsentiert. Ansonsten liegt die Anzahl der Knoten des top path in  $[\log(\log(n)), 3\log(\log(n))]$ . Der bottom tree ist ein Hilfsbaum vergleichbar mit denen von Tango-Baum. Der Operationssatz unterscheidet sich jedoch etwas. Für die Höhe  $h$  der hier verwendeten Hilfsbäume, muss aber auch  $h = O(\log p)$  gelten, mit  $p$  ist die Anzahl der im Hilfsbaum enthaltenen Knoten.

Die  $l$  Knoten im top path enthalten so repräsentieren diese die Knoten aus dem Pfad  $p_1, p_2, \dots, p_l$ . Der top path wird in zig und zag Sequenzen unterteilt. zig Sequenzen entsprechen den längst möglichen Teilpfaden von Knoten mit linken Kindern. zag Sequenzen entsprechen den längst möglichen Teilpfaden von Knoten mit rechten Kindern. In Abbildung 1 sind zig und zag Sequenzen dargestellt. Ist  $v_z$  der letzte Knoten einer zig bzw. zag Sequenz, dann müssen in der zugehörigen preferred Path Repräsentation, die Knoten mit größerer Tiefe als  $v_z$  größere bzw. kleinere Schlüssel als *key* ( $v_z$ ) haben. Der bottom tree aus Abbildung 1 kann also nur Schlüssel aus (50, 60) enthalten.



**Abbildung 1:** zig Sequenzen sind grün dargestellt. zag Sequenzen blau

## Literatur

- [1] Prosenjit Bose, Karim Douïeb, Vida Dujmović, and Rolf Fagerberg. An  $o(\log \log n)$ -competitive binary search tree with optimal worst-case access times. *Algorithm Theory - SWAT 2010*, page 38–49, 2010.
- [2] Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. The landscape of bounds for binary search trees. *CoRR*, abs/1603.04892, 2016.