

Titel (vorläufig):**TANGO BÄUME**

Suchbäume gehören zu den bekanntesten und wichtigsten Datenstrukturen in der Informatik. Viele verschiedene Varianten wurden für unterschiedliche Aufgaben entwickelt und über die Zeit verbessert. Tango Bäume [1] wurden entwickelt, um möglichst effizient mit verschiedenen Abfolgen von Anfragen umzugehen. Es wurde gezeigt, dass in dieser Datenstruktur für jeden mögliche Abfolgen von Anfragen die Laufzeit maximal um einen Faktor $O(\log \log n)$ schlechter ist, als die beste Laufzeit, die mit Suchbäumen (mit bestimmten Eigenschaften) überhaupt erreicht werden kann. Es wird vermutet, dass Splay-Bäume nur um einen konstanten Faktor $O(1)$ von der besten Laufzeit abweichen. Diese „Dynamische Optimalitäts Vermutung“ konnte bis heute jedoch nicht bewiesen werden.

In dieser Arbeit sollen Tangobäume vorgestellt, implementiert und getestet werden.

Aufgaben:

Literaturrecherche: Stellen Sie verschiedene Arten von Suchbäumen vor. Gehen Sie insbesondere auf solche Varianten ein, die gute Laufzeitgarantien gegenüber der optimalen Laufzeit garantieren (Splay Bäume, Tango Bäume, Multisplay Bäume, Zipper Bäume). Vergleichen Sie die bekannten Laufzeitgarantien und diskutieren Sie die „Dynamische Optimalitäts Vermutung“. Beweisen Sie die $O(\log \log n)$ -Kompetitivität von Tango-Bäumen in eigenen Worten. Erklären Sie zudem die sogenannte „interleave lower bound“ von Wilber [2] und den Zusammenhang mit Tango Bäumen.

Implementierung: Implementieren Sie die Datenstruktur der Tango Bäume. Die Implementierung sollte so angelegt sein, dass die intern verwendeten balancierten Suchbäume (im original Rot-Schwarz Bäume) auch durch eine andere Datenstruktur ersetzt werden können. Eine graphische Darstellung der internen Abläufe ist optional.

Implementieren Sie zudem Splay Bäume (die Find-Operation genügt) und vergleichen Sie die Laufzeiten der beiden Datenstrukturen. Nutzen Sie für diesen Vergleich verschiedene Arten von Abfolgen von Anfragen (zufällig generierte und nach eigenen Mustern erstellte).

Literatur (Auswahl):

- [1] E. D. Demaine, D. Harmon, J. Iacono, and M. Patrascu. Dynamic optimality - almost. *SIAM J. Comput.*, 37(1):240–251, 2007.
- [2] R. E. Wilber. Lower bounds for accessing binary search trees with rotations. *SIAM J. Comput.*, 18(1):56–67, 1989.