Aufgabe 3.1

T. Adam, M. ben Ahmed

Universität Osnabrück

Æ

December 7, 2020

Aufgabe 3.1 - Übersicht

Testen von Suchbaum Layouts

- zufällig
- sortiert
- levelweise
- van Emde Boas

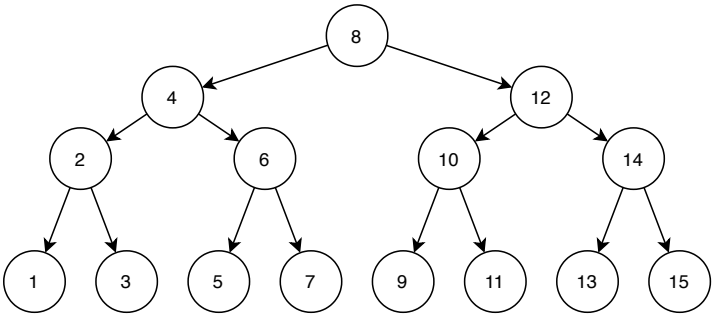
https://staff.fnwi.uva.nl/p.vanemdeboas/

Aufgabe 3.1 - (Perfekt) balancierter binärer Suchbaum

Baumstruktur

- Perfekt balancierter binärer Suchbaum
- linker Kindknoten $<$ Elternknoten $<$ rechter Kindknoten
- n Level
- $2^n - 1$ Schlüssel
- Integerarray speichert die Schlüssel

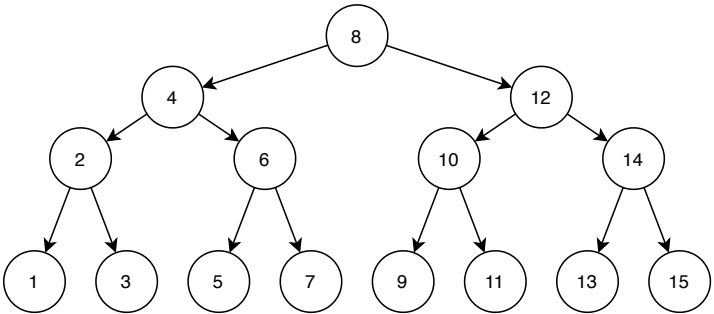
Aufgabe 3.1 - Zufällig



5	11	14	4	10	12	1	13	7	15	2	9	11	6	3

`find(key)`: iteriere über array

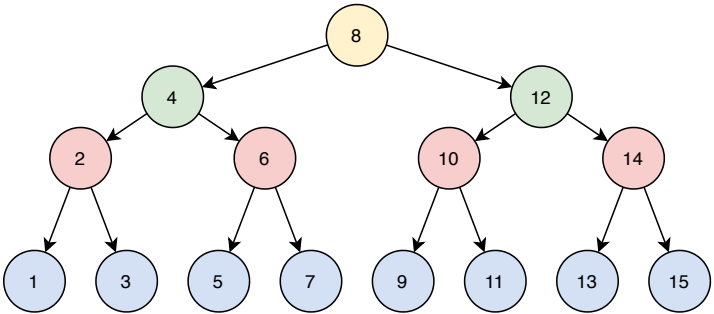
Aufgabe 3.1 - Aufsteigend sortiert



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

`find(key)`: binäre Suche

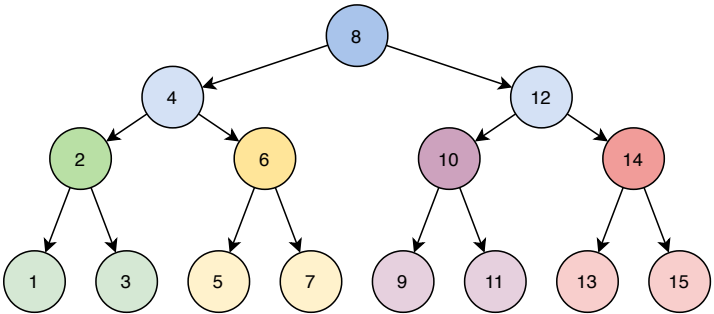
Aufgabe 3.1 - Levelweise sortiert



8	4	12	2	6	10	14	1	3	5	7	9	11	13	15

`find(key)`: $links(i) = 2i$, $rechts(i) = 2i + 1$

Aufgabe 3.1 - van Emde Boas Layout



8	4	12	2	1	3	6	5	7	10	9	11	14	13	15

`find(key)`: ???

Aufgabe 3.1 - Berechnung der Indizes im VEB Layout

Van Emde Boas Layout

- wir betrachten zwei mögliche Implementationen
- es ist kein Konstantzeitalgorithmus für Indizeberechnung bekannt
- Konvertieren von `bfs`-Index zu VEB-Index in $\mathcal{O}(\log h)$
- alternative Implementation durch Zeiger auf Kindknoten

Aufgabe 3.1 - Testmaschinen

Intel Core i5-3570K

- 4c/4t
- 3,4 - 3,8 GHz
- 6 MB L3
- 8 GB (1600 MHz)
- Linux (Kubuntu 18.04)

Intel Core i7-3770K

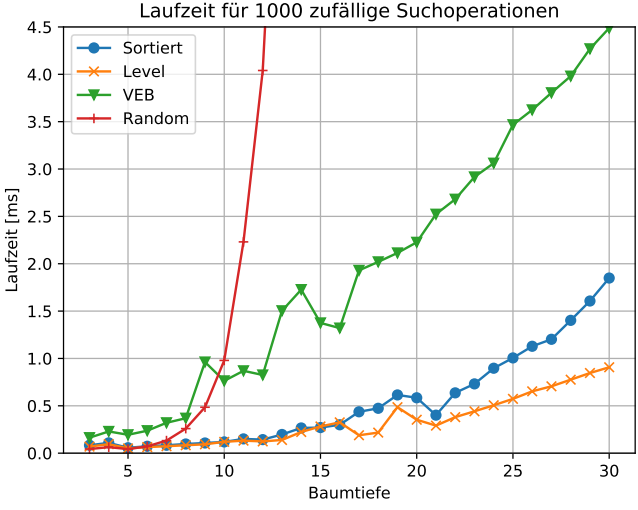
- 4c/8t
- 3,5 - 3,9 GHz
- 8 MB L3
- 16 GB (1600 MHz)
- Linux (Kubuntu 18.04)

Intel Core i7-8565U

- 4c/8t
- 1,8 - 4,6 GHz
- 8MB L3
- 16 GB (2400 MHz)
- Linux (Arch)

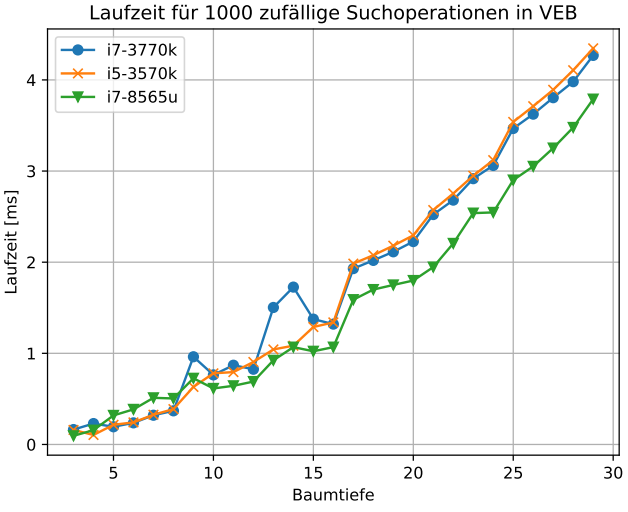
Aufgabe 3.1 - Auswertung

Vergleich der Layouts



Aufgabe 3.1 - Auswertung

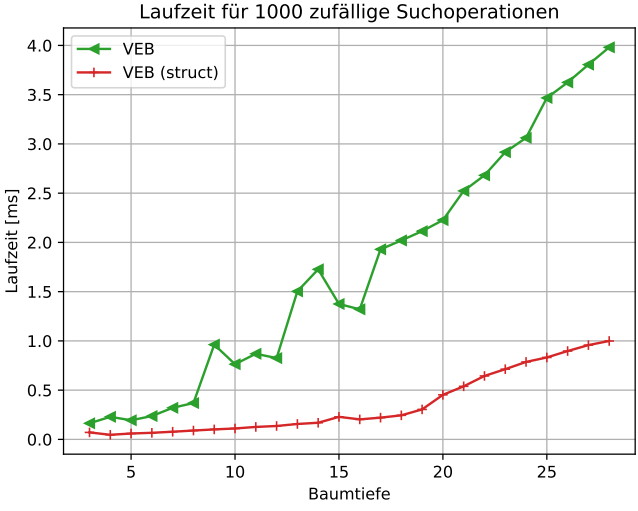
Vergleich der Maschinen



Aufgabe 3.1 - Auswertung

Vergleich der VEB Implementationen

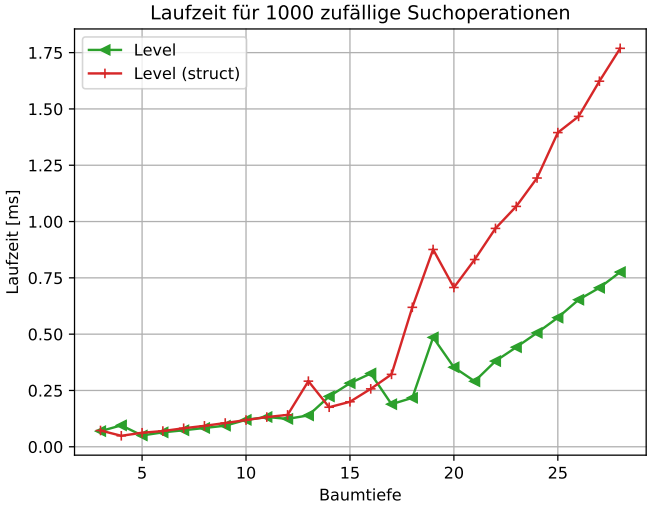
- Berechnen von Indizes von entfällt



Aufgabe 3.1 - Auswertung

Vergleich der Level Implementationen

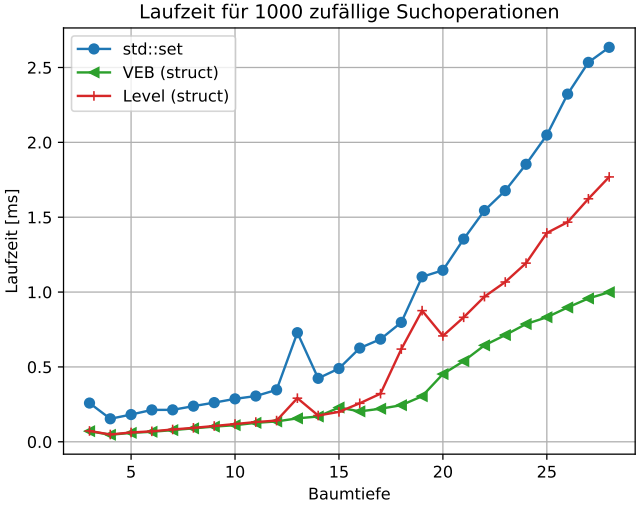
- struct-Implementation verbraucht mindestens den **dreifachen** Speicher



Aufgabe 3.1 - Auswertung

Vergleich zur STL Implementation

- Die struct Implementationen von VEB und Level sind schneller als `std::set`



Aufgabe 3.1 - Auswertung

Vergleich einiger Layout Implementationen und `std::set`