

Aufgabe 10

Sei B ein binärer Suchbaum. In jedem Knoten v von B wird ein Schlüssel $v.key \in \mathbb{N}$ gespeichert sowie Zeiger $v.left$, $v.right$ und $v.parent$ auf sein linkes Kind, auf sein rechtes Kind und auf seinen Elternknoten. Die Zeiger sind nil , wenn der entsprechende Nachbar nicht existiert. Für zwei Knoten u und v ist wie üblich der *Abstand* die Anzahl der Kanten auf dem kürzesten Pfad von u nach v .

Für einen Knoten w von B sei $B(w)$ der Teilbaum von B mit Wurzel w . Für zwei Knoten u und v von B ist w ein *gemeinsamer Vorfahre*, wenn u und v in $B(w)$ liegen. Wir suchen den niedrigsten gemeinsamen Vorfahren $ngV(u, v)$ von u und v , also einen gemeinsamen Vorfahren w , so dass für jeden Vorfahren w von u und v gilt, dass w in $B(w)$ liegt. Wir betrachten verschiedene Szenarien, in denen Sie jeweils den niedrigsten gemeinsamen Vorfahren von u und v berechnen sollen.

Exkurs: Lowest Common Ancestor

Als Lowest Common Ancestor (LCA) oder „letzter gemeinsamer Vorfahre“ wird in der Informatik und Graphentheorie ein Ermittlungskonzept bezeichnet, das einen gegebenen gewurzelten Baum von Datenstrukturen effizient vorverarbeitet, sodass anschließend Anfragen nach dem letzten gemeinsamen Vorfahren für beliebige Knotenpaare in konstanter Zeit beantwortet werden können. ^a

^ahttps://de.wikipedia.org/wiki/Lowest_Common_Ancestor

- (a) Wir bekommen u und v als Zeiger auf die entsprechenden Knoten in B geliefert. Beschreiben Sie in Worten und in Pseudocode einen Algorithmus, der den niedrigsten gemeinsamen Vorfahren von u und v berechnet. Analysieren Sie die Laufzeit Ihres Algorithmus.
- (b) Wir bekommen u und v wieder als Zeiger auf die entsprechenden Knoten in B geliefert. Seien d_u , und d_v , die Abstände von u bzw. v zum niedrigsten gemeinsamen Vorfahren von u und v . Die Laufzeit Ihres Algorithmus soll $O(\max\{d_u, d_v\})$ sein. Dabei kann Ihr Algorithmus in jedem Knoten v eine Information $v.info$ speichern. Skizzieren Sie Ihren Algorithmus in Worten.
- (c) Wir bekommen die Schlüssel $u.key$ und $v.key$. Die Laufzeit Ihres Algorithmus soll proportional zum Abstand der Wurzel von B zum niedrigsten gemeinsamen Vorfahren von u und v sein. Skizzieren Sie Ihren Algorithmus in Worten.