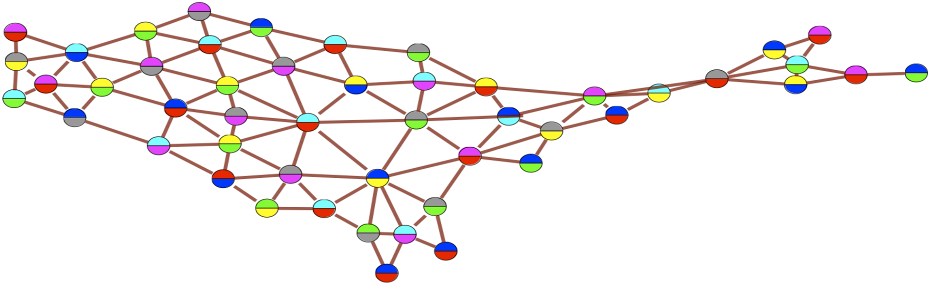
Karlsruher Institut für Technologie

Tutorium 8: Sortierte Folgen

Holger Ebhart | 10. Juni 2015

TUTORIUM ZUR VORLESUNG ALGORITHMEN I IM SS15



Gliederung

Sortierte Folgen

Wir wollen eine sortierte Folge verwalten und dabei sortiert halten.
Folgende Funktionen sollen unterstützt werden:

- locate(e)
- insert
- remove

Laufzeit von $\mathcal{O}(\log n)$ soll erreicht werden.
$\Rightarrow$ Baumstruktur (Suchbaum)

Implementierung

- speichere binären Baum
- Blätter sind die Elemente
- Knoten sind Spaltschlüssel mit Schlüssel s für die gilt:
  - $\forall e \in Elemente$ mit $e \leq s$: e ist links von s
  - $\forall e \in Elemente$ mit $e > s$: e ist rechts von s

Implementierung

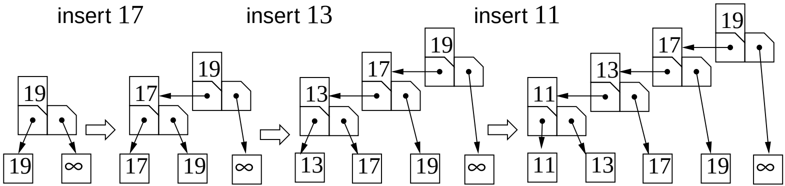
Karlsruher Institut für Technologie

- speichere binären Baum
- Blätter sind die Elemente
- Knoten sind Spaltschlüssel mit Schlüssel *s* für die gilt:
  - $\forall e \in \mathit{Elemente}$ mit $e \leq s$: e ist links von s
  - $\forall e \in \mathit{Elemente}$ mit $e > s$: e ist rechts von s

Beispiel 1

Beispiel 2

Beispiel



Problem: Der Baum wird beliebig unbalanciert.

$\rightsquigarrow$ langsam

(a,b)-Bäume

Nun Versuch die Bäume besser zu balancieren.
Dazu einen binär Baum durch einen (a,b)-Baum zu ersetzen.

- node hat nun maximal b Zeiger
- und mindestens a Zeiger (außer Wurzel)

$\Rightarrow$ Baumhöhe $\approx \log_a(n)$

- nach einem split müssen gültige Knoten entstehen $\rightarrow \lfloor \frac{b+1}{2} \rfloor \geq a$
- nach einem fuse müssen gültige Knoten entstehen
  $\rightarrow a + (a-1) \leq b$

$\Rightarrow$ Bedingung: $a \geq 2$ und $b \geq 2a - 1$

- locate ?

(a,b)-Bäume

Nun Versuch die Bäume besser zu balancieren.
Dazu einen binär Baum durch einen (a,b)-Baum zu ersetzen.

- node hat nun maximal b Zeiger
- und mindestens a Zeiger (außer Wurzel)

$\Rightarrow$ Baumhöhe $\approx \log_a(n)$

- nach einem split müssen gültige Knoten entstehen $\rightarrow \lfloor \frac{b+1}{2} \rfloor \geq a$
- nach einem fuse müssen gültige Knoten entstehen $\rightarrow a + (a-1) \leq b$

$\Rightarrow$ Bedingung: $a \geq 2$ und $b \geq 2a - 1$

- locate ?

(a,b)-Bäume

Karlsruher Institut für Technologie

Nun Versuch die Bäume besser zu balancieren.
Dazu einen binär Baum durch einen (a,b)-Baum zu ersetzen.

- node hat nun maximal b Zeiger
- und mindestens a Zeiger (außer Wurzel)

$\Rightarrow$ Baumhöhe $\approx \log_a(n)$

- nach einem split müssen gültige Knoten entstehen $\rightarrow \lfloor \frac{b+1}{2} \rfloor \geq a$
- nach einem fuse müssen gültige Knoten entstehen
  $\rightarrow a + (a-1) \leq b$

$\Rightarrow$ Bedingung: $a \geq 2$ und $b \geq 2a - 1$

- locate ?

(a,b)-Bäume

Nun Versuch die Bäume besser zu balancieren.
Dazu einen binär Baum durch einen (a,b)-Baum zu ersetzen.

- node hat nun maximal b Zeiger
- und mindestens a Zeiger (außer Wurzel)

$\Rightarrow$ Baumhöhe $\approx \log_a(n)$

- nach einem split müssen gültige Knoten entstehen $\rightarrow \lfloor \frac{b+1}{2} \rfloor \geq a$
- nach einem fuse müssen gültige Knoten entstehen
  $\rightarrow a + (a-1) \leq b$

$\Rightarrow$ Bedingung: $a \geq 2$ und $b \geq 2a - 1$

- locate ?

(a,b)-Bäume

Nun Versuch die Bäume besser zu balancieren.
Dazu einen binär Baum durch einen (a,b)-Baum zu ersetzen.

- node hat nun maximal b Zeiger
- und mindestens a Zeiger (außer Wurzel)

$\Rightarrow$ Baumhöhe $\approx \log_a(n)$

- nach einem split müssen gültige Knoten entstehen $\rightarrow \lfloor \frac{b+1}{2} \rfloor \geq a$
- nach einem fuse müssen gültige Knoten entstehen
  $\rightarrow a + (a-1) \leq b$

$\Rightarrow$ Bedingung: $a \geq 2$ und $b \geq 2a - 1$

- locate ?

(a,b)-Bäume - Beispiel

(a,b)-Bäume - insert

```
procedure insert(e: Element)
    find path root-next Element e'
    insertBefore(e,e')
    insert key(e) as splitter in parent u
    if size(u) = b+1 then
        split(u)

procedure split(n:node)
    split n in 2 nodes (⌊(b+1)/2⌋, ⌈(b+1)/2⌉)
    insert new splitter in parent u
    if u is root then return
    if size(u) = b+1 then
        split(u)
```

(a,b)-Bäume - delete

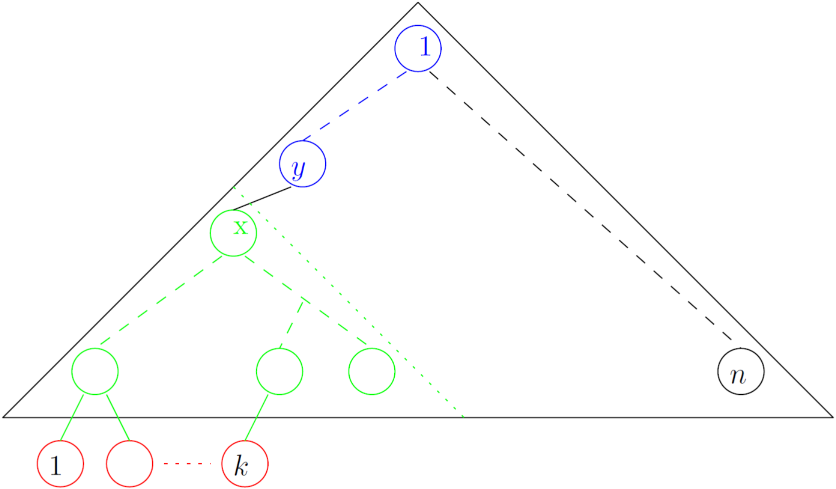

```
procedure remove(e: Element)
    find path root e
    delete(e)
    remove splitter key(e) in parent u
    if size(u) = a-1 then
        merge(u, neighbour u')

procedure merge(n,m:node)
    if size(m) + a -1 ≤ b then
        u := fuse(n,m)
        if u is root then return
        merge(u, neighbour u')
    else balance(u,u')
```

(a,b)-Bäume - Beispiel insert und delete

Bulk Insertion on Heaps

Es sei ein binärer Heap aus n Elementen gegeben. Nun sollen k weitere Elemente auf einmal eingefügt werden. Geben Sie ein Verfahren (kein Pseudocode) an, mit dem man das Einfügen in $\mathcal{O}(\min\{k \log k + \log n, k + \log n \log k\})$ Schritten erledigen kann.
Sie können davon ausgehen, dass der Heap genau $2^m - 1$ Elemente enthält ($m \in \mathbb{N}$).

Bulk Insertion on Heaps - Lösungsidee



Pancake-Sort

Gegeben seien n Pancakes in unterschiedlicher Größe auf einem Stapel. Man hat nun einen Pancake-Flipper zur Verfügung mit dem man die obersten l ($l \leq n, l \in \mathbb{N}$) Pancakes umdrehen kann. Entwickeln Sie einen schnellen Algorithmus der die Pancakes sortiert.

Zu Übungsblatt 7

Sortieren Sie folgende Menge mit Radixsort (dezimal, d=3):
$Z = (111, 76, 223, 567, 349, 496, 201, 872, 3)$
Geben Sie alle Zwischenschritte an.

Zu Übungsblatt 7

Eine k-Clique $Q = (V', E')$ des Graphen $G = (V, E)$ ist ein vollständiger Teilgraph von G für den gilt:
$V' \subseteq V, E' \subseteq E, E' = V' \times V'$ und $|V| = k$

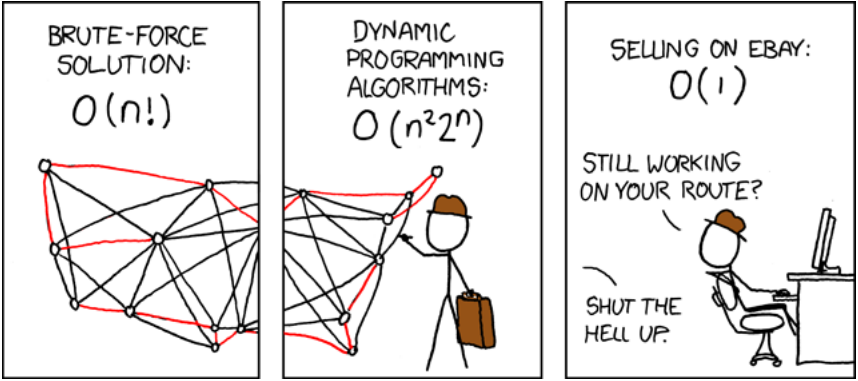
In welchem der folgenden Graphen gibt es eine 4-Clique? Geben Sie die Knoten jeder 4-Clique an.Graphen siehe Tafel!

Zu Übungsblatt 7

Eine k-Clique $Q = (V', E')$ des Graphen $G = (V, E)$ ist ein vollständiger Teilgraph von G für den gilt:
$V' \subseteq V, E' \subseteq E, E' = V' \times V'$ und $|V| = k$

In welchem der folgenden Graphen gibt es eine 4-Clique? Geben Sie die Knoten jeder 4-Clique an.Graphen siehe Tafel!

Vielen Dank für eure Aufmerksamkeit!
Bis zum nächsten Mal.



stackoverflow.com