Adaptive Huffman Komprimierung

Niklas Jugl, Moritz Bruder

16. Mai 2013

Inhaltsverzeichnis

Überlegungen

Überlegungen

- Es handelt sich nicht um einen (binären) Suchbaum

Überlegungen

- Es handelt sich nicht um einen (binären) Suchbaum
- Suche von Knoten über die Breite ist effizienter, da häufige Knoten weiter oben sind

Überlegungen

- Es handelt sich nicht um einen (binären) Suchbaum
- Suche von Knoten über die Breite ist effizienter, da häufige Knoten weiter oben sind
- Knotennummern steigen nach oben und rechts an

Umsetzung

Umsetzung

- Deshalb gilt für jeden Knoten:

Umsetzung

- Deshalb gilt für jeden Knoten:

```
nodes[node.number] == node
```

Umsetzung

- Deshalb gilt für jeden Knoten:

```
nodes[node.number] == node
```

- Array wächst von oben herunter

Problem

Addressierung findet auf Byteebene statt.

Problem

Addressierung findet auf Byteebene statt.

Lösung

Problem

Addressierung findet auf Byteebene statt.

Lösung

- 1 Byte zwischenspeichern

Problem

Addressierung findet auf Byteebene statt.

Lösung

- 1 Byte zwischenspeichern
- Manipuliere mit Bitoperatoren zum Lesen und Schreiben

1. Baum absteigen (links wenn 0 gelesen, rechts sonst)

1. Baum absteigen (links wenn 0 gelesen, rechts sonst)
2. Knoten mit Symbol gefunden? Gebe Symbol aus

1. Baum absteigen (links wenn 0 gelesen, rechts sonst)
2. Knoten mit Symbol gefunden? Gebe Symbol aus
3. Ansonsten empfange Symbol

1. Lineare Suche nach Knoten mit Symbol im Array

1. Lineare Suche nach Knoten mit Symbol im Array
2. Wenn nicht gefunden, dann handelt es sich um NYT, hänge Symbol umgekehrt an

1. Lineare Suche nach Knoten mit Symbol im Array
2. Wenn nicht gefunden, dann handelt es sich um NYT, hänge Symbol umgekehrt an
3. Wandere den Baum hoch, wenn momentaner Knoten linkes Kind war hänge 0 an, sonst 1

1. Lineare Suche nach Knoten mit Symbol im Array
2. Wenn nicht gefunden, dann handelt es sich um NYT, hänge Symbol umgekehrt an
3. Wandere den Baum hoch, wenn momentaner Knoten linkes Kind war hänge 0 an, sonst 1
4. Drehe Kodierung um

Bereits in der Vorlesung besprochen.

Bereits in der Vorlesung besprochen.

Blockprinzip

Ein Block besteht aus allen Knoten mit gleichem Gewicht.

Bereits in der Vorlesung besprochen.

Blockprinzip

Ein Block besteht aus allen Knoten mit gleichem Gewicht.

Bereits in der Vorlesung besprochen.

Blockprinzip

Ein Block besteht aus allen Knoten mit gleichem Gewicht.

1. Wird Knoten zu Symbol nicht gefunden hänge einen neuen ein

Bereits in der Vorlesung besprochen.

Blockprinzip

Ein Block besteht aus allen Knoten mit gleichem Gewicht.

1. Wird Knoten zu Symbol nicht gefunden hänge einen neuen ein
2. Wandere den Baum nach oben und wenn der aktuelle Knoten nicht der mit der größten Nummer im Block ist, tausche diesen mit dem größten aus

Bereits in der Vorlesung besprochen.

Blockprinzip

Ein Block besteht aus allen Knoten mit gleichem Gewicht.

1. Wird Knoten zu Symbol nicht gefunden hänge einen neuen ein
2. Wandere den Baum nach oben und wenn der aktuelle Knoten nicht der mit der größten Nummer im Block ist, tausche diesen mit dem größten aus
3. Erhöhe dann das Gewicht

- Binaries

- Binaries
- Zu finden auf github.com unter *AdaptiveHuffmanCompression*

- Binaries
- Zu finden auf github.com unter *AdaptiveHuffmanCompression*
- Calgary corpus