

# Übung 13 – Algorithmen II

Moritz Laupichler, Hans-Peter Lehmann – {moritz.laupichler, hans-peter.lehmann}@kit.edu  
[http://algo2.iti.kit.edu/AlgorithmenII\\_WS22.php](http://algo2.iti.kit.edu/AlgorithmenII_WS22.php)

Institut für Theoretische Informatik - Algorithmik II

```
    result = current_weight;
    return true;
}

for( EdgeID eid = graph.edgeBegin( current ); eid != graph.edgeEnd( current ); ++eid ){
    const Edge & edge = graph.getEdge( eid );
    COUNTING( statistic_data.inc( DijkstraStatisticData::TOUCHED_EDGES ); )
    if( edge.forward ){
        COUNTING( statistic_data.inc( DijkstraStatisticData::RELAXED_EDGES ); )
        weight new_weight = edge.weight + current_weight;
        GUARANTEE( new_weight >= current_weight, std::runtime_error, "Weight overflow detected." );
        if( !priority_queue.isReached( edge.target ) ){
            COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_EDGES ); )
            COUNTING( statistic_data.inc( DijkstraStatisticData::REACHED_NODES ); )
            priority_queue.push( edge.target, new_weight );
        } else {
            if( priority_queue.getCurrentKey( edge.target ) > new_weight ){
                COUNTING( statistic_data.inc( DijkstraStatisticData::SUCCESSFULLY_RELAXED_NODES ); )
                priority_queue.decreaseKey( edge.target, new_weight );
            }
        }
    }
}
```

- Stringology
  - Huffman Codes
- Blatt 05, A4
- Blatt 05, A6

# Huffman-Kodierung

## Wiederholung

- Zähle Vorkommen jedes Zeichens
- Verbinde jeweils die zwei Knoten mit geringstem Gewicht
- Beschrifte die Kanten des Baums mit 0/1
- Präfixfreie Codewörter ergeben sich durch Pfad im Baum

# Huffman-Kodierung

## Beispiel

$T =$  karlsruherinstitutfuertechnologie

$T =$

{e}: 4 {u}: 3 {o}: 2 {k}: 1 {l}: 2 {s}: 2 {h}: 2 {n}: 2 {a}: 1 {f}: 1 {c}: 1 {g}: 1 {r}: 4 {i}: 3 {t}: 4 {e}:

# Huffman-Kodierung

## Beispiel

Zeichen	Codewort
e	000
i	110
r	101
t	111
h	0110
l	0100
n	0111
s	0101
u	0010
a	10000
c	10010
f	10001
g	10011
k	00111
o	00110

- Alle Codewörter mit gleicher Länge haben aufsteigende Codes
- Innerhalb der Codes sind die Zeichen alphabetisch sortiert
- Berechnung:
  - Code-Längen bestimmen mit normalem Huffman
  - Sortieren nach Länge und Zeichen
  - Binär durchnummerieren (ohne leading zero)
  - Bei größerer Länge nach dem Inkrementieren von rechts mit Nullen auffüllen

# Kanonische Huffman-Kodierung

## Beispiel

Zeichen	Codewort	Kanonisches Codewort
e	000	000
i	110	001
r	101	010
t	111	011
h	0110	1000
l	0100	1001
n	0111	1010
s	0101	1011
u	0010	1100
a	10000	11010
c	10010	11011
f	10001	11100
g	10011	11101
k	00111	11110
o	00110	11111

# Kanonische Huffman-Kodierung

## Platzeffiziente Repräsentation

- Tabelle mit jeweils letztem Codewort
- Dann Index innerhalb der Zeile durch Subtraktion bestimmbar

Letztes Codewort	Zeichentabelle
011	e, i, r, t
1100	h, l, n, s, u
11111	a, c, f, g, k, o

- Beispiel: Code = 111001111111111111**111** 001111111111111**110**  
011111111111111**1100** 1111111111  
(3+1)-tes Zeichen von hinten → f





# Ende!



# Feierabend!