

# DISKRETE STRUKTUREN

## PRAKTIKA 1

---

Liu Kin

Wombacher Sascha

24. April 2016

1. Allgemeines
2. Rank
3. Unrank
4. Successor und Predecessor

## ALLGEMEINES

---

Allgemeines:

- Aufgabe 1 ist lediglich eine Festlegung von  $q = 2$
- rank/unrank/successor werden für allgemeine  $q$  erklärt
- Wenn  $q$  einen geraden Wert besitzt ist es auch ein zyklischer Code

Anmerkung:

Rekursionsgleichung wurde aus dem Skript übernommen (möglicherweise ein Schreibfehler in der Aufgabenstellung!)

**RANK**



Allgemeines:

- Dient zur Bestimmung des “Index” eines übergebenen Codes
- Benötigt wissen über das Alphabet  $q$
- Benötigt wissen über die Länge des Codes

Algorithmus:

- Beginne von rechts
- Gelesenes Symbol beschreibt einen Bereich
- Dieser Bereich beschreibt ein Offset für den Rückgabe Index
- Rekursiver Aufruf ohne des Gelesen Symbols
- Rückgabe bei Geradem Symbol (“Leserichtung von links”):  
Offset + RekursiverRückgabewert
- Rückgabe bei Ungeradem Symbol (“Leserichtung von rechts”):  
Offset - RekursiverRückgabewert - 1 + <BereichGröße>

Übergebenes Objekt:  $rank(012)$ ,  $q = 3$

Lese letztes Element: 2 (gerade)

Rückgabe:  $2 \cdot q^2 + rank(01) = 23$

Lese letztes Element: 1 (ungerade)

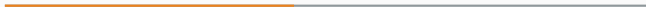
Rückgabe:  $1 \cdot q^1 - rank(0) - 1 + q^1 = 5$

Lese letztes Element: 0 (gerade, kein folge Element)

Rückgabe: 0



UNRANK



Allgemeines:

- Dient zur Bestimmung eines Codes anhand eines “Index”
- Benötigt wissen über das Alphabet  $q$
- Benötigt wissen über die Länge des Codes

Algorithmus:

- Beginne von rechts das Array zu füllen
- Generiere  $q$  Bereiche über  $q^{bitPosition}$  Werte
- Schreibe die Bereich Position an die nächste Array Position
- Passe den Index (für rekursion) an
- Ist der Bereich ungerade “invertiere” alle nachfolge Werte  
invertiere:  $Wert = q - 1 - Wert$
- rekursiver Aufruf, bis alle Array Positionen besetzt sind

Übergebener Index:  $i = 23$ ,  $q = 3$ ,  $length = 3$

Bereiche:  $0 - 8$ ,  $9 - 17$ ,  $18 - 26$

Schreibe 2 (gerade, keine Anpassungen);  $unrank(23 - 18 = 5, \text{nicht inventieren})$

Bereiche:  $0 - 2$ ,  $3 - 5$ ,  $6 - 8$

Schreibe 1 (ungerade, inventiere folge Rekursionen);  $unrank(5 - 3 = 2, \text{inventiere})$

Bereiche:  $0, 1, 2$

Eigentliche Rückgabe: 2

inventieren:  $3 - 1 - 2 = 0$

## SUCCESSOR UND PREDECESSOR

---

Allgemeines:

- Dienen zur Bestimmung des nächsten (vorherigen) Codes
- Benötigt wissen über das Alphabet  $q$
- (Benötigt wissen über die Länge des Codes)

Algorithmus:

- Hole den Index des Objektes mit der *Rank-Funktion*
- Addiere/Subtrahiere 1 von dem erhaltenen Index
- Generier ein neues Objekt mit der *Unrank-Funktion*

Performanzproblem:

- Generierung eines neuen Objektes  
=> vorzüge von *minimal-change* gehen “verloren”

Algorithmus:

- Rufe Ink-/Dek-rementiere auf letztes Element auf
- Wiederhole dies bis zum ersten Element  
Bei Ungeraden Eigenwert: inventiere den Aufruf  
(ink wird zu dek und umgekehrt)
- Kann das erste Element nicht in/de krementieren  
Versuche das zuvor kommende Element



## SUCCESSOR UND PREDECESSOR

### PART 4: BEISPIEL 1 - SUCCESSOR

Übergebenes Objekt:  $\text{succ}(012)$ ,  $q = 3$

$\text{ink}(012) \ 2 \rightarrow \text{ink}(01)$

$\text{ink}(01) \ 1 \rightarrow \text{dek}(0)$

$\text{dek}(0)$  nicht möglich  $\rightarrow$  return false

$\text{ink} \ 1$  möglich  $\rightarrow$  schreibe 2, return true

Ergebnis:

022

## SUCCESSOR UND PREDECESSOR

### PART 4: BEISPIEL 2 - PREDECESSOR

Übergebenes Objekt:  $pred(012)$ ,  $q = 3$

$dek(012) \ 2 \rightarrow dek(01)$

$dek(01) \ 1 \rightarrow ink(0)$

$ink(0)$  möglich  $\rightarrow$  schreibe 1, return true

Ergebnis:

112

HABEN SIE FRAGEN?



**VIELEN DANK FÜR DIE AUFMERKSAMKEIT!**

---