

# Enhanced and Extended Suffix Arrays

Adrian Regenfuß

Technische Universität München

Sommersemester 2020

# Übersicht

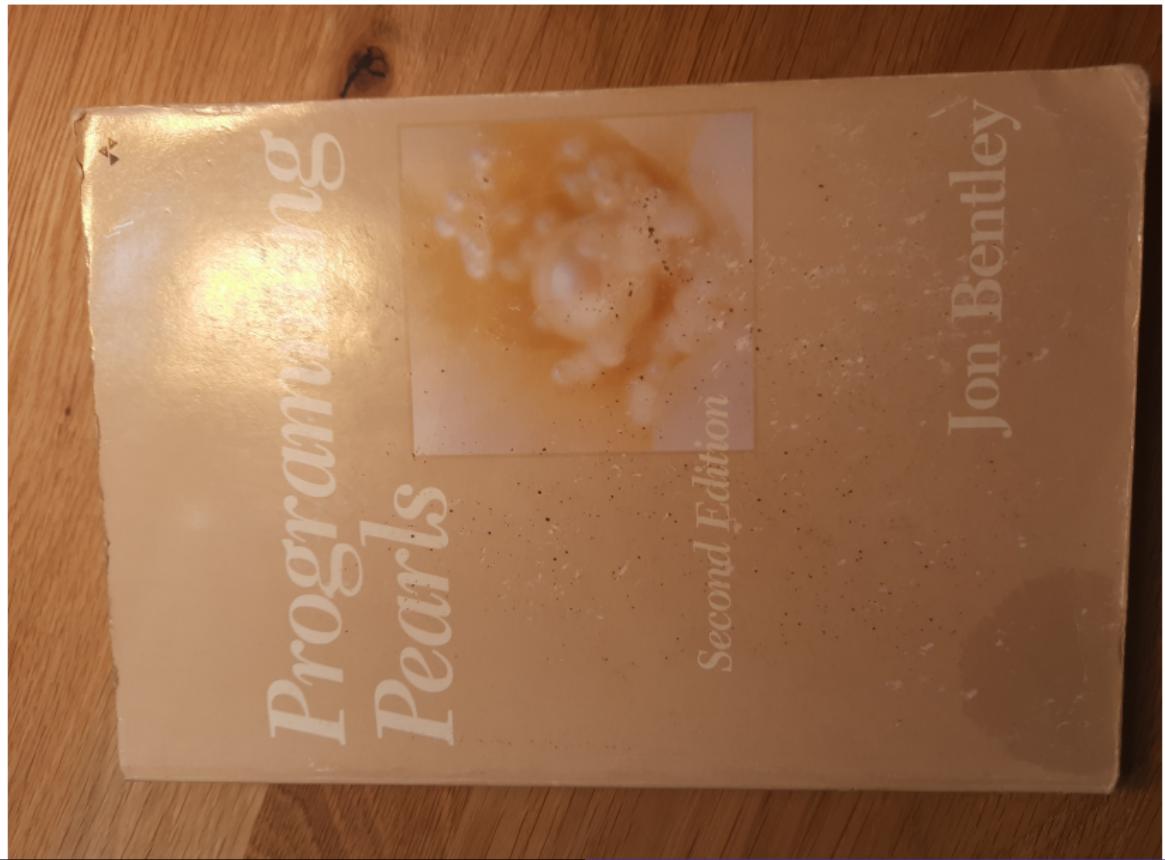
1 A Personal Anecdote

2 Structure

4 years ago

- Me in Nepal
- No computer with me, only pen, paper and Programming Pearls

# My old, damaged copy of Programming Pearls



# Thinking a lot about Algorithms

2017-04-16-01  
at interpretation of  $\beta$  from  $\beta[0..n-1]$  to  $\text{int\_lin\_inter\_find}(\beta[], n, k)$

$\beta[0..n-1] \in \mathbb{Z}$   
 $n = \text{int}(\beta[0..n-1])$

• else if  $\beta[n-1] \leq k$        $i = (\beta[n-1] - \beta[0]) / (k - \beta[0]) + 1$   
ret  $i$

$\ell = 0$

$v = n - 1$

loop

$m = \text{ceil}((\beta[0..v] - \beta[0]) / (k - \beta[0])) + \ell$

if  $\beta[m] \geq k \& \& \beta[m-1] \leq k$

ret  $m$

else if  $\beta[m] < k$

$\ell = m$

else  $v = m - 1$

while  $\ell < v$

$m = \text{ceil}((\beta[0..v] - \beta[0]) / (k - \beta[0])) + \ell$

if  $\beta[m] \geq k \& \& \beta[m-1] \leq k$

ret  $m$

else if  $\beta[m] < k$

$\ell = m$

else  $v = m - 1$

ret  $v$

if  $\text{lin\_find}(\beta[], n, k) \neq \text{int\_lin\_find}(\beta[], n, k)$

for  $i = 0; i < n \& \& \beta[i] < k; i++$

ret  $i$

# Problems back then

- Thinking about searching, sorting, collision detection, string matching
- Specifically: What is the longest repeated substring of a string?
- Or: what is the longest supermaximal repeat of a string?
- Devising very complex algorithms

# Discovering Suffix Arrays

- On the flight back reading the 15th chapter of Programming Pearls
- Describes this exact problem, solved using suffix arrays

# Übersicht

1 A Personal Anecdote

2 Structure

# Structure

- Description of data structures
- Clarification of terminology
- Description of algorithms

# Übersicht

1 A Personal Anecdote

2 Structure

# Suffix Array