Text Indexing

Lecture 00: Course Overview

Florian Kurpicz
Organizational Matters

Lectures
- Monday 10:00–11:30 (50.34, -119)
- lecture only
## Organizational Matters

### Lectures
- Monday 10:00–11:30 (50.34, -119)
- lecture only

### Project (mandatory)
- topics will be handed out 08.11.2021
- coding project and small presentation
- 20% of the final grade
# Organizational Matters

## Lectures
- Monday 10:00–11:30 (50.34, -119)
- lecture only

## Project (mandatory)
- topics will be handed out 08.11.2021
- coding project and small presentation
- 20% of the final grade

## Oral Exam
- 20 minutes
- 80% of the final grade
- pizza marks content not relevant for exam
Organizational Matters

Lectures
- Monday 10:00–11:30 (50.34, -119)
- lecture only

Project (mandatory)
- topics will be handed out 08.11.2021
- coding project and small presentation
- 20 % of the final grade

Oral Exam
- 20 minutes
- 80 % of the final grade
- pizza marks content not relevant for exam

Office Hours (Room 210)
- Monday 13:45–14:45 (lecture period)
- by appointment (otherwise)
Materials

**Slides**
- published after the lecture
  (https://algo2.iti.kit.edu/4198.php)

**Videos**
- will be published (with $\geq 1$ week delay)
Materials

**Slides**
- published after the lecture
  (https://algo2.iti.kit.edu/4198.php)

**Videos**
- will be published (with $\geq$ 1 week delay)

**Additional Material**
- references to literature included
- books
- most likely no script
## Content

### Fundamentals
- tries
- suffix tree
- suffix array
- longest common prefix array
- Burrows-Wheeler transform (BWT)
- wavelet tree (+ bit vector rank/select)
- FM-index

### Compressed Indices
- compressing the BWT and wavelet trees
- Lempel-Ziv 77/78 compression
- LZ compression vs. BWT compression
- compressed suffix trees and suffix arrays
- r-index

### Additional Topics
- parallel construction
- different query types
Motivation for Text Indices

- collection of text
- scanning not feasible
## Motivation for Text Indices

<table>
<thead>
<tr>
<th>word</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1, #3, #7, ...</td>
<td>#1, #3, #7, ...</td>
</tr>
<tr>
<td>#2, #3, ...</td>
<td>#2, #3, ...</td>
</tr>
<tr>
<td>#2, #4, #5, ...</td>
<td>#2, #4, #5, ...</td>
</tr>
<tr>
<td>#1, #2, ...</td>
<td>#1, #2, ...</td>
</tr>
<tr>
<td>#3, #7, #9, ...</td>
<td>#3, #7, #9, ...</td>
</tr>
<tr>
<td>#4, #5, #6, ...</td>
<td>#4, #5, #6, ...</td>
</tr>
</tbody>
</table>

- collection of text
- scanning not feasible
- inverted index (word based)
Motivation for Text Indices

- collection of text
- scanning not feasible
- inverted index (word based)
Motivation for Text Indices

- collection of text
- scanning not feasible
- inverted index (word based)
Motivation for Text Indices

- collection of text
- scanning not feasible
- inverted index (word based)
- phrase search
Motivation for Text Indices

- collection of text
- scanning not feasible
- inverted index (word based)
- phrase search
- counting queries
Motivation for Text Indices

- collection of text
- scanning not feasible
- inverted index (word based)
- phrase search
- counting queries
- what if there are no “words”

<table>
<thead>
<tr>
<th>word</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1, #3, #7, ...</td>
</tr>
<tr>
<td></td>
<td>#2, #3, ...</td>
</tr>
<tr>
<td></td>
<td>#2, #4, #5, ...</td>
</tr>
<tr>
<td></td>
<td>#1, #2, ...</td>
</tr>
<tr>
<td></td>
<td>#3, #7, #9, ...</td>
</tr>
<tr>
<td></td>
<td>#4, #5, #6, ...</td>
</tr>
<tr>
<td></td>
<td>:</td>
</tr>
</tbody>
</table>

GAATGCCAGTCAGCATTAGGCCAGGCA
GGAGAGCTCAGGGCAGGTCACGTGGGA
AACTCGCATAGTGAGGTTATCGCTCG
ACATGGTCGTGGGCTCTCACCTCCTT
CCGACACGAACTCGATTTAGTITGTAT
CTACATCTACAGAGGTTGGCAGCTTA
TGTCGCCCGTGTAGAGGAGGAAAGG
TCGGAATTTCGATTTTCAAGAGCTCGGA
CTCGTCACTTTCCAGAATACGAAT
CATCGCCTGCAAGC AAAATGGAATAG
GACGTTTAAATGGAACCCTGGACATTCG
AATCGCATGTAGGTTATCGGGA
ACATGGTCGTGGGCTCTCACCTCCTT
CCGACACGAACTCGATTTAGTITGTAT
TGTCGCCCGTGTAGAGGAGGAAAGG
CTACATCTACAGAGGTTGGCAGCTTA
CATCGCCTGCAAGC AAAATGGAATAG
Why Texts?

Text is Everywhere

- Text-based Information
  - Wikipedia
  - dblp
  - books
  - news articles
  - code
- Very Important in Bioinformatics
  - DNA
  - proteins

Growth of DNA Sequencing

- Recorded growth
- Double every 7 months (Historical growth rate)
- Double every 12 months (Illumina Estimate)
- Double every 18 months (Moore’s Law)

[Ste+15]
Definition: Text

- let $\Sigma$ be an alphabet
- $T \in \Sigma^*$ is a text
- $|T| = n$ is the length of the string
### Definition: Text
- let $\Sigma$ be an **alphabet**
- $T \in \Sigma^*$ is a text
- $|T| = n$ is the length of the string

### Definition: Alphabet Types
- **constant size alphabet**: finite set not depending on $n$
- **integer alphabet**: alphabet is $\{1, \ldots, \sigma\}$ and fits into constant number of words
- **finite alphabets**: alphabet of finite size
Definition: Substring, Prefix, and Suffix


- $T[i..j] = T[i] \ldots T[j]$ is called a **substring**, $a \ b \ b \ a \ a \ b \ b \ a \ $
Definition: Substring, Prefix, and Suffix


- $T[i..j] = T[i] \ldots T[j]$ is called a **substring**,
  
  
  | a | b | b | a | a | b | b | a |

- $T[1..i]$ is called a **prefix**, and
  
  | a | b | b | a | a | b | b | a |

**Sentinel for Simplicity**

Given a text $T$ of length $n$ over an alphabet $\Sigma$, we assume that $T[n] = \$ \in \Sigma$ and $\$ < $\alpha$ for all $\alpha \in \Sigma$ otherwise, suffix can be prefix of another suffix.

**Definition: Prefix-Free**

A string is **prefix-free** if no suffix is a prefix of another suffix.
Definition: Substring, Prefix, and Suffix

- $T[i..j] = T[i] \ldots T[j]$ is called a substring,
  
  \[
  \text{a b b a a b b a $}
  \]

- $T[1..i]$ is called a prefix, and
  
  \[
  \text{a b b a a b b a $}
  \]

- $T[i..n]$ is called a suffix of $T$.
  
  \[
  \text{a b b a a b b a $}
  \]
Definition: Substring, Prefix, and Suffix


- $T[i..j] = T[i] \ldots T[j]$ is called a substring,

```
  a b b a a b b a $
```

- $T[1..i]$ is called a prefix, and

```
  a b b a a b b a $
```

- $T[i..n]$ is called a suffix of $T$.

```
  a b b a a b b a $
```

Sentinel for Simplicity

Given a text $T$ of length $n$ over an alphabet $\Sigma$.

- we assume that $T[n] = $ with

```
  a b b a a b b a $
```

- $\not\in \Sigma$ and $< \alpha$ for all $\alpha \in \Sigma$
Definition: Substring, Prefix, and Suffix

Given a text \( T = T[1]T[2] \ldots T[n] \) of length \( n \):
- \( T[i..j] = T[i] \ldots T[j] \) is called a substring,
- \( T[1..i] \) is called a prefix, and
- \( T[i..n] \) is called a suffix of \( T \).

Sentinel for Simplicity

Given a text \( T \) of length \( n \) over an alphabet \( \Sigma \).
- we assume that \( T[n] = $ \) with
- \( $ \notin \Sigma \) and \( $ < \alpha \) for all \( \alpha \in \Sigma \)
Definition: Substring, Prefix, and Suffix
- $T[i..j] = T[i] \ldots T[j]$ is called a substring,
  $a\ b\ b\ a\ a\ b\ b\ a\ $ $\$
- $T[1..i]$ is called a prefix, and
  $a\ b\ b\ a\ a\ b\ b\ a\ $ $\$
- $T[i..n]$ is called a suffix of $T$.
  $a\ b\ b\ a\ a\ b\ b\ a\ $ $\$

Sentinel for Simplicity
Given a text $T$ of length $n$ over an alphabet $\Sigma$:
- we assume that $T[n] = \$ $ with
  $\$ \notin \Sigma$ and $\$ < $ for all $\alpha \in \Sigma$
- otherwise, suffix can be prefix of another suffix

$T[1..n] = abbaabba$ and $T[5..n] = abba$
Definition: Substring, Prefix, and Suffix

Given a text \( T = T[1] \ldots T[n] \) of length \( n \):
- \( T[i..j] = T[i] \ldots T[j] \) is called a **substring**,
  \[
  \text{a b b a a b b a $}
  \]
- \( T[1..i] \) is called a **prefix**, and
  \[
  \text{a b b a a b b a $}
  \]
- \( T[i..n] \) is called a **suffix** of \( T \).
  \[
  \text{a b b a a b b a $}
  \]

Sentinel for Simplicity

Given a text \( T \) of length \( n \) over an alphabet \( \Sigma \).
- we assume that \( T[n] = $ \) with
  \[
  \text{$ / \in \Sigma \text{ and$ < \alpha \text{ for all } \alpha \in \Sigma} \]
- otherwise, suffix can be prefix of another suffix
  \[
  \begin{array}{cccccccc}
  1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
  \text{a b b a a b b a} \\
  \end{array}
  \]
  \[
  \begin{array}{cccccccc}
  T[1..n] = \text{abbaabba and } T[5..n] = \text{abba} \\
  \end{array}
  \]

Definition: Prefix-Free

A string is **prefix-free** if no suffix is a prefix of another suffix.
PINGO
Bibliography

