

Sorting by Length-Weighted Reversals: Dealing with Signs and Circularity

Firas Swidan

Michael A. Bender

Dongdong Ge

Simai He

Haodong Hu

Ron Y. Pinter

Background

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

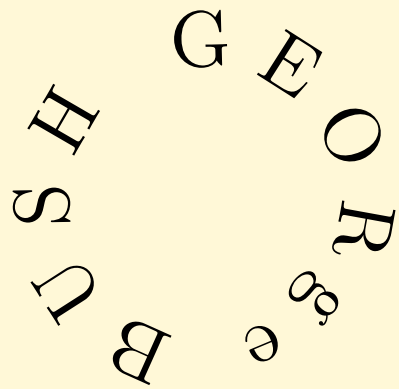
Results

Sorting 0/1 Sequences

[courtesy of Michael Bender]

The Reversal Problem

Transfer one circular sequence into the other by reversals:



A circular arrangement of letters: G, E, O, R, e, U, S, H.

[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

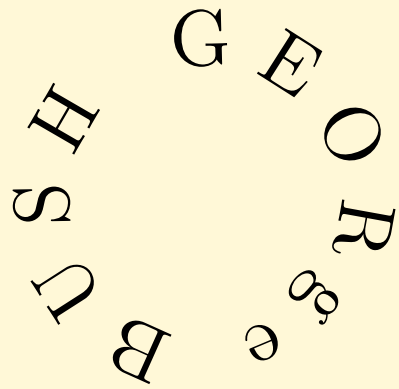
Aspects of the Problem

Results

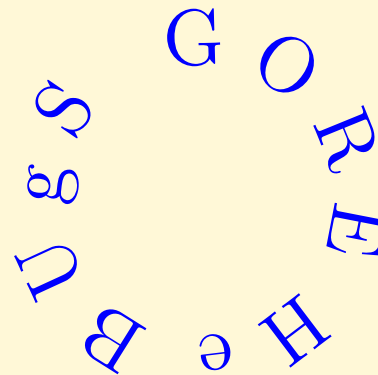
Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



A circular sequence of letters: G, E, O, R, e, B, U, S, H. The letters are arranged in a circle, with 'G' at the top and 'H' at the bottom.



A circular sequence of letters: G, O, R, E, H, e, B, U, S. The letters are arranged in a circle, with 'G' at the top and 'S' at the bottom.

[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

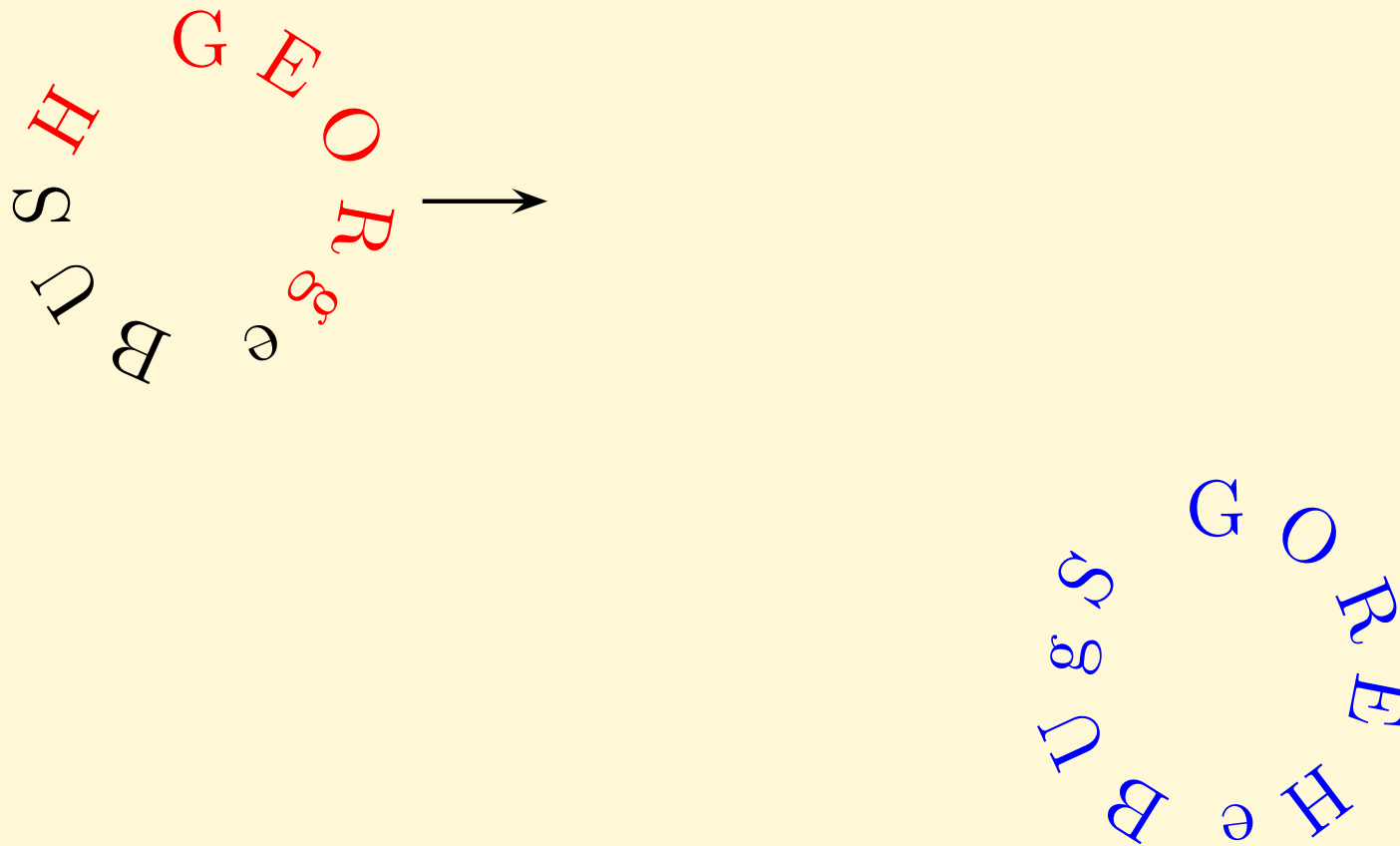
Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

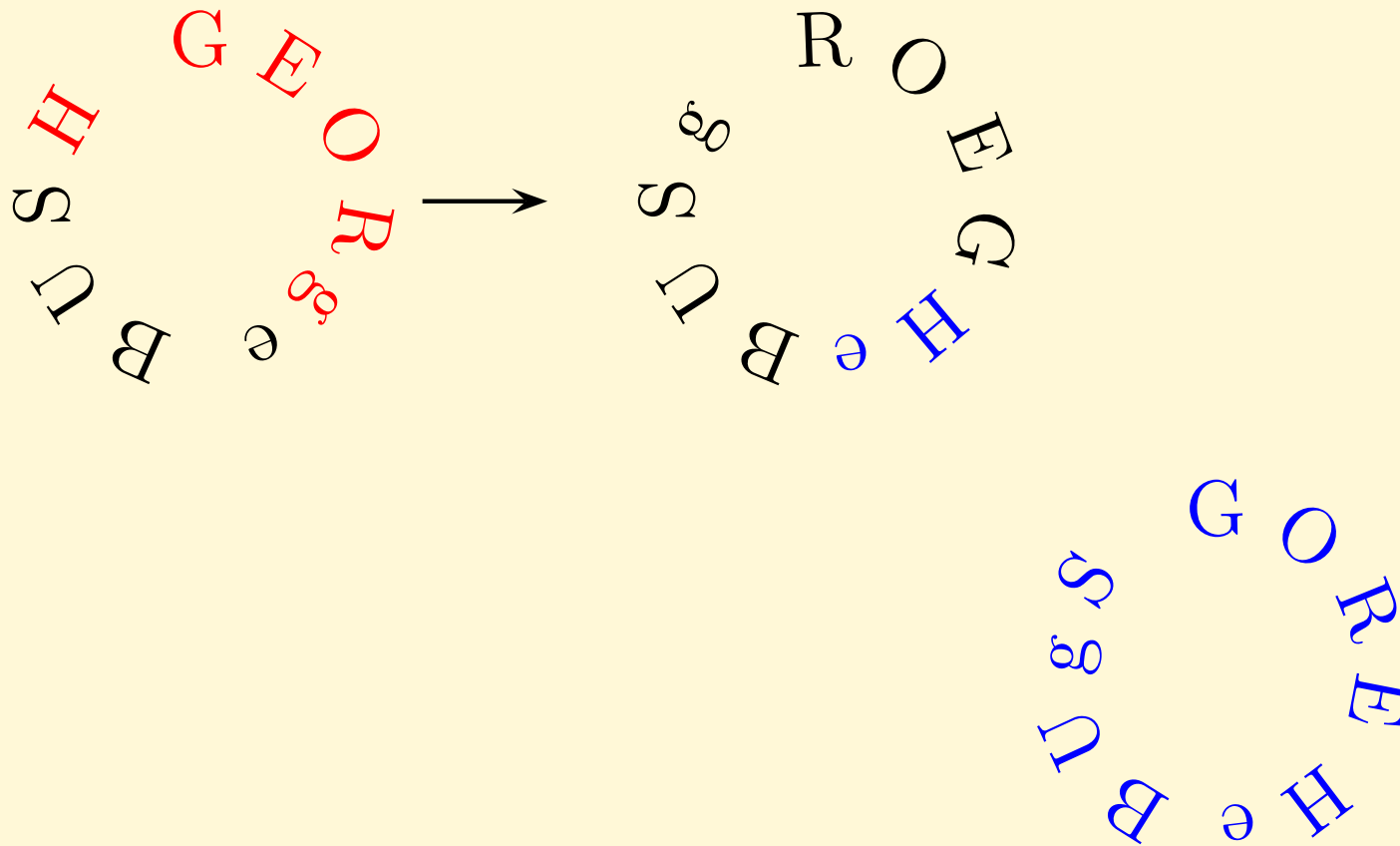
Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

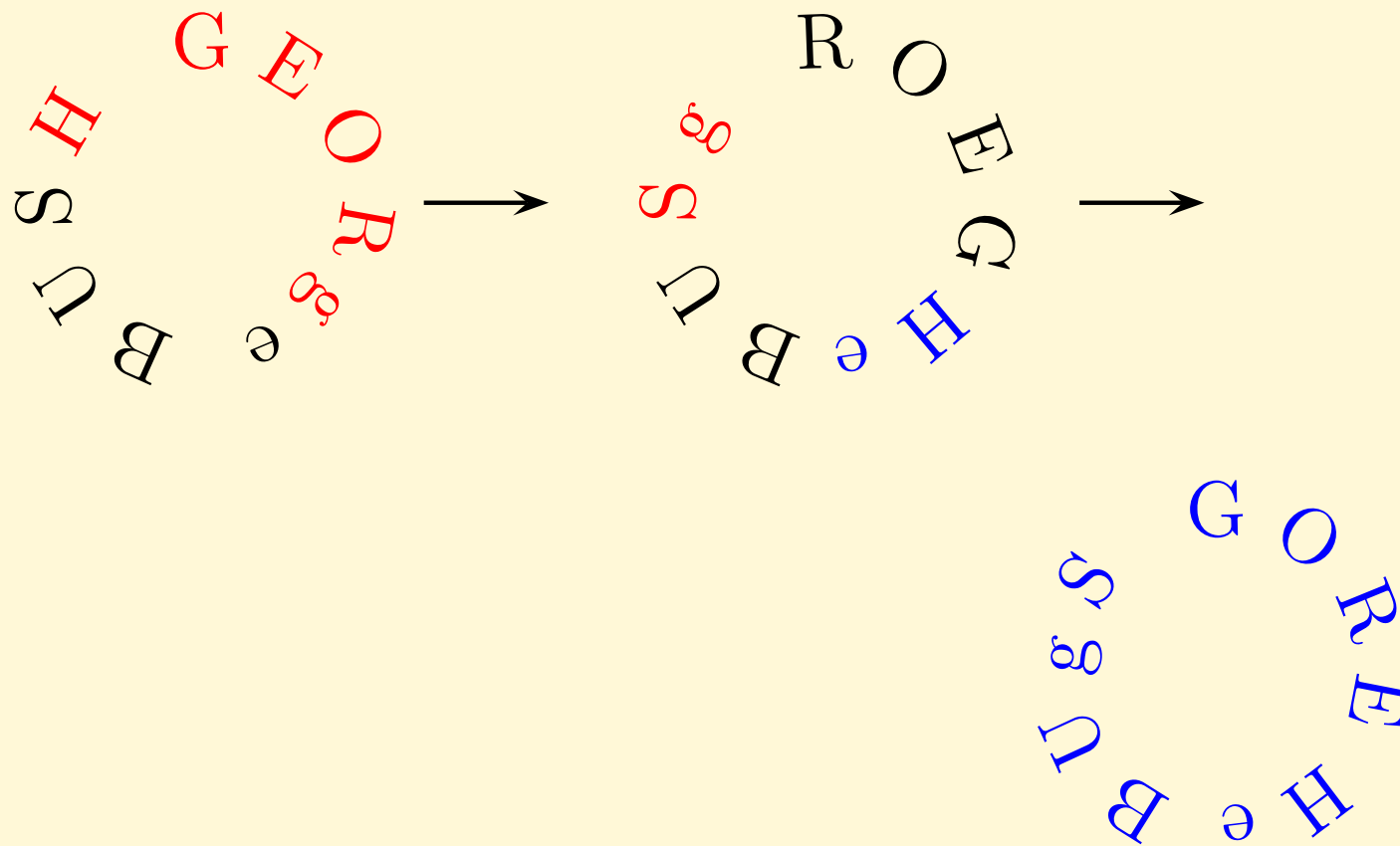
Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

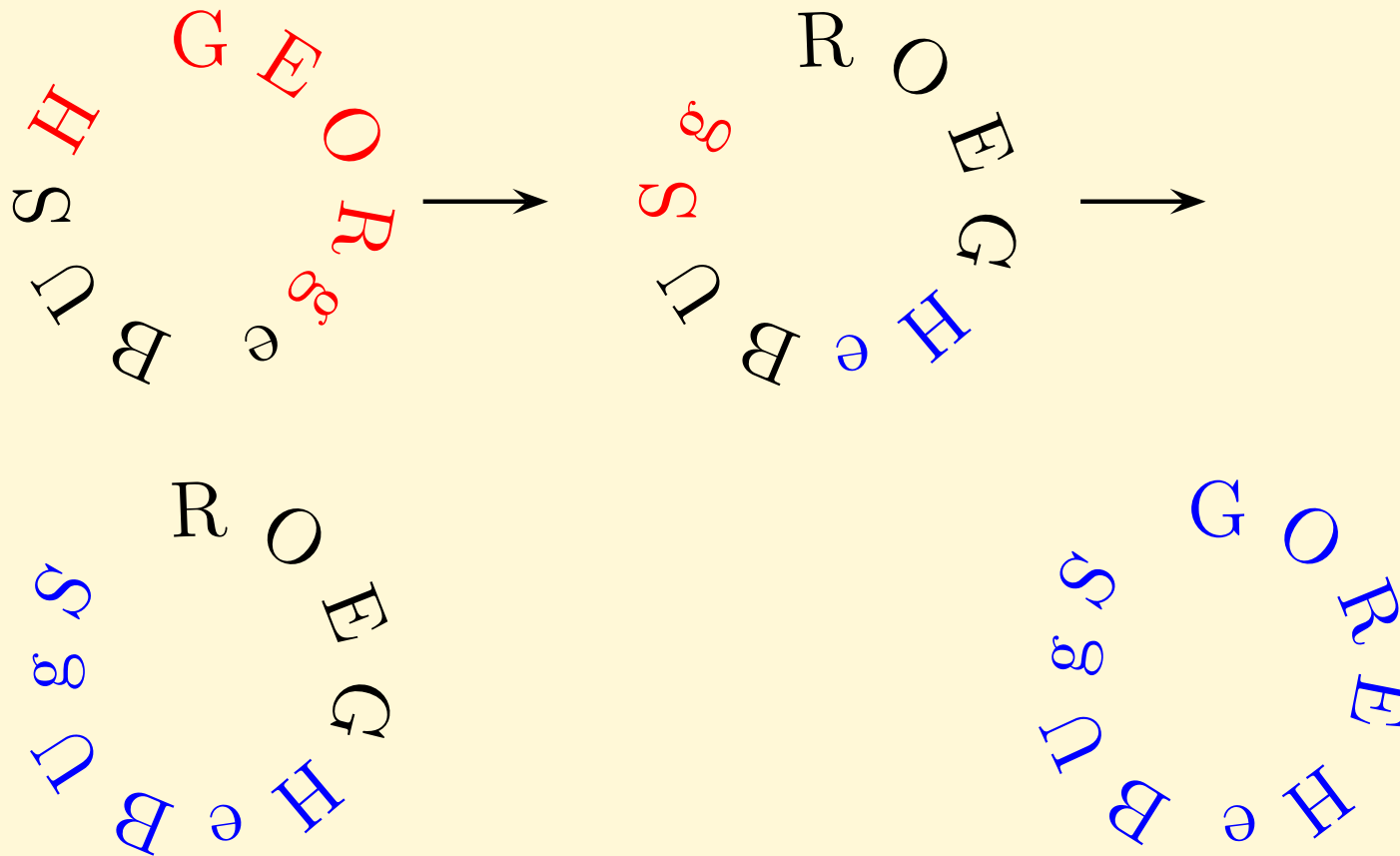
Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

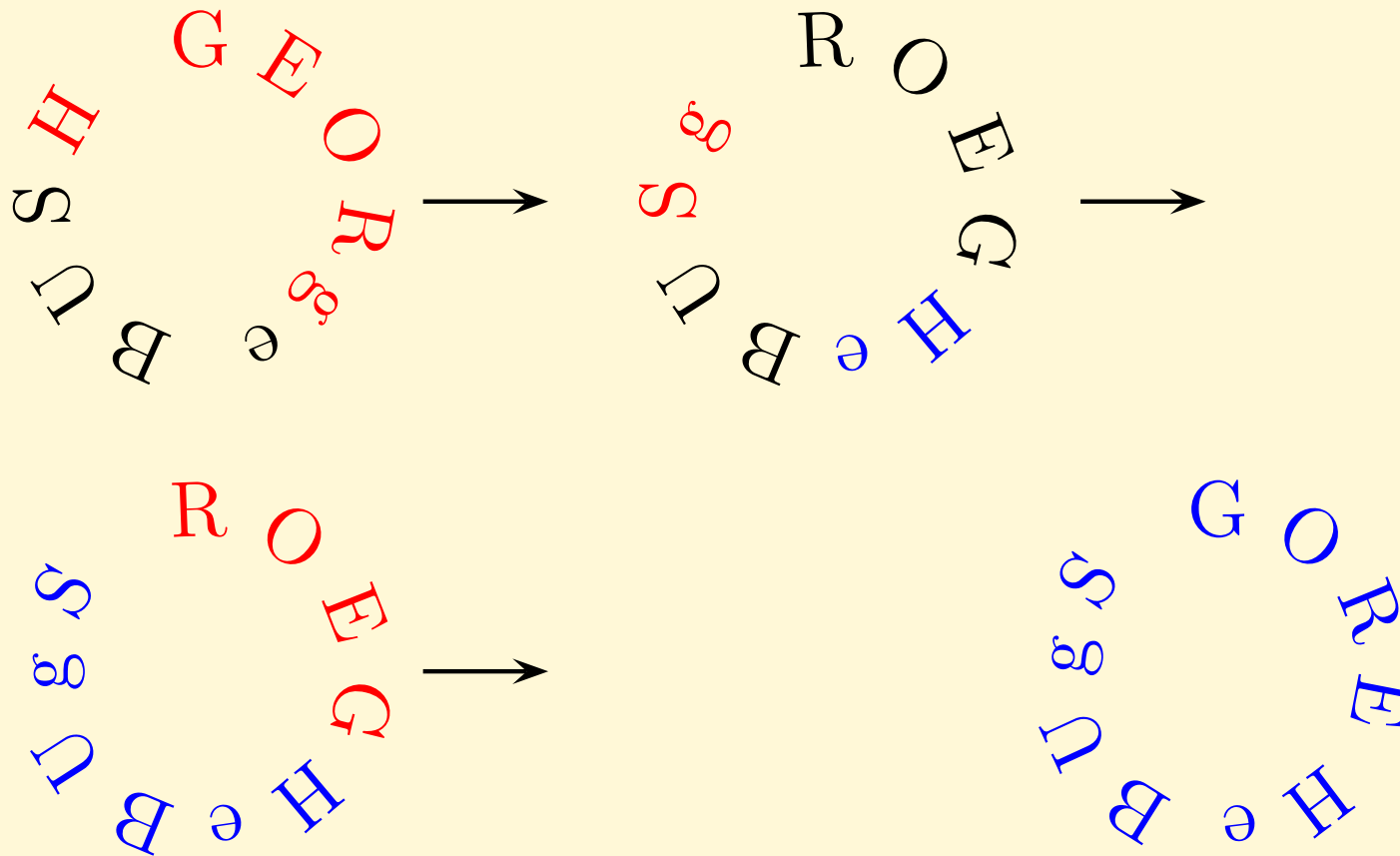
Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

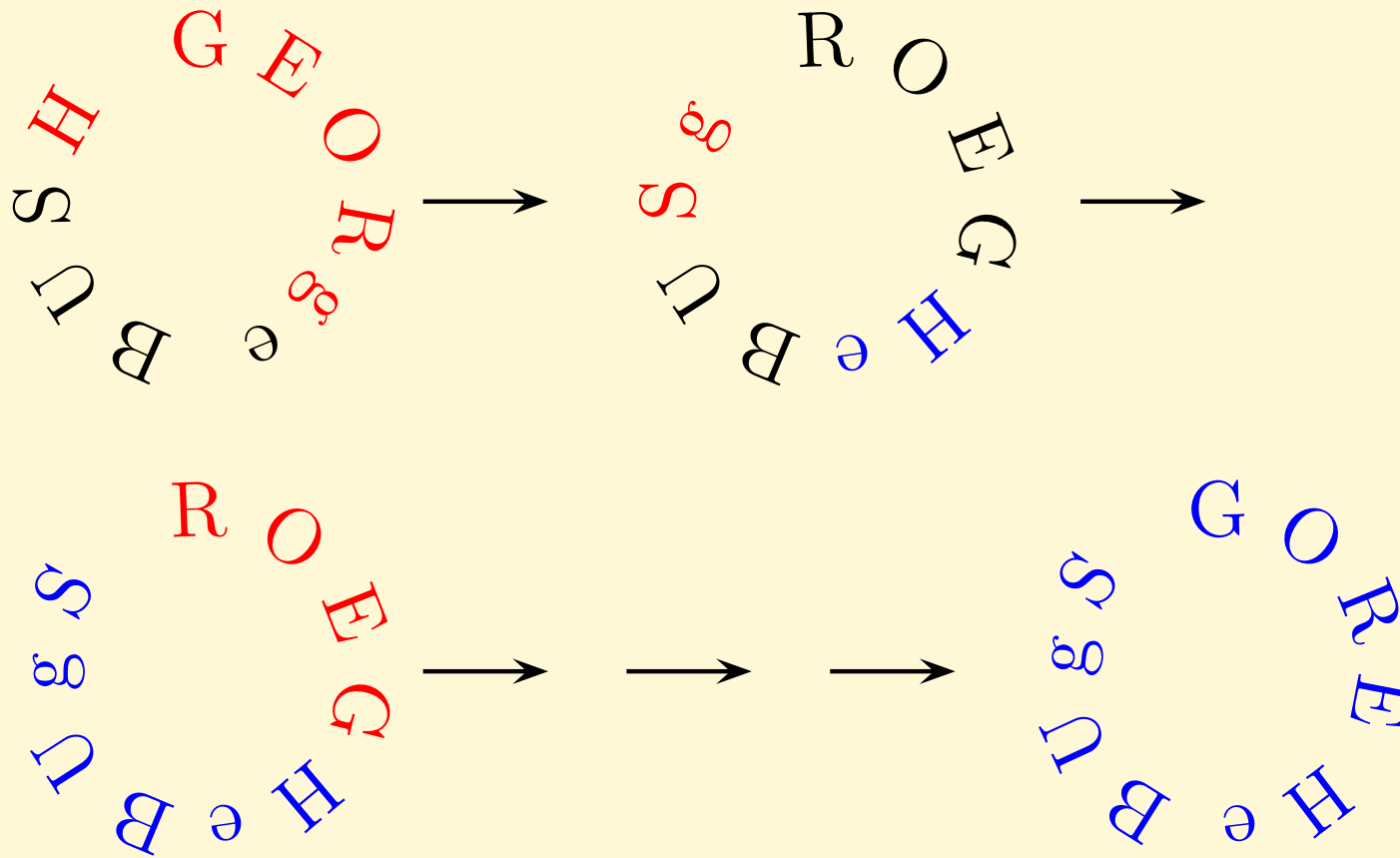
Aspects of the Problem

Results

Sorting 0/1 Sequences

The Reversal Problem

Transfer one circular sequence into the other by reversals:



[courtesy of Michael Bender]

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

Sorting by Reversals

When there are no duplications, rename the sequences into permutations.

[Background](#)

[The Reversal Problem](#)

[Sorting by Reversals](#)

[Applications](#)

[0/1 Sorting](#)

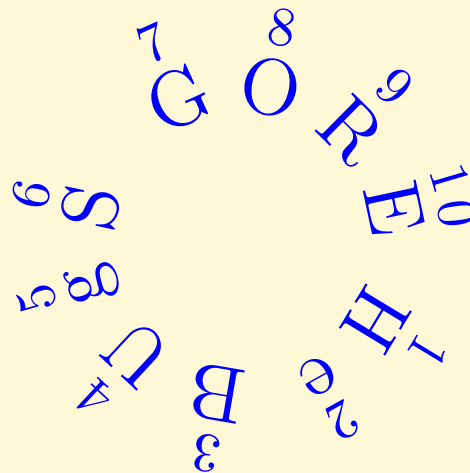
[Aspects of the Problem](#)

[Results](#)

[Sorting 0/1 Sequences](#)

Sorting by Reversals

When there are no duplications, rename the sequences into permutations.



Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

Sorting by Reversals

When there are no duplications, rename the sequences into permutations.

Background

The Reversal Problem

Sorting by Reversals

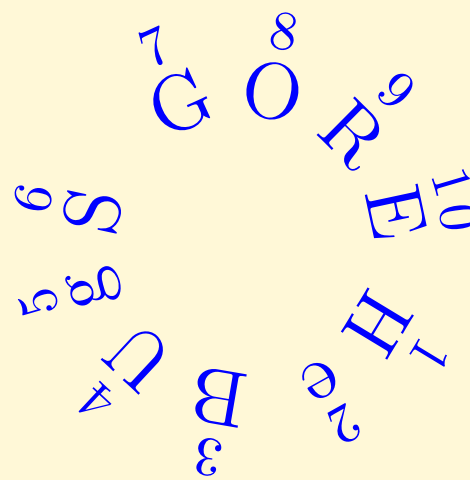
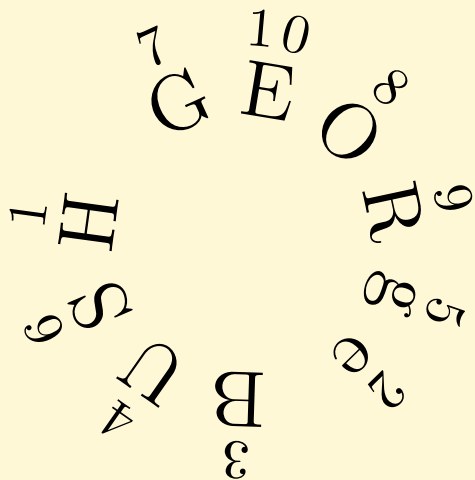
Applications

0/1 Sorting

Aspects of the Problem

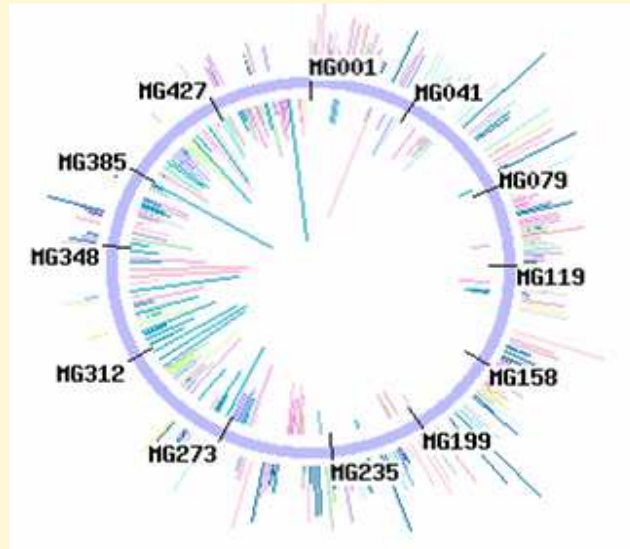
Results

Sorting 0/1 Sequences



- Comp. genomics: seq. = genomes, chars. = genes, reversals = mutations.
- Min-cost reversal dist: evolutionary dist.
- Opt solution: likely evolutionary path.

Figure 1: Genome of *Mycoplasma genitalium*



Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

- Another extreme: as many duplications as makes sense.

[Background](#)

[The Reversal Problem](#)

[Sorting by Reversals](#)

[Applications](#)

[0/1 Sorting](#)

[Aspects of the Problem](#)

[Results](#)

[Sorting 0/1 Sequences](#)

- Another extreme: as many duplications as makes sense.
- Resembles genomes when considering them as a sequence of nucleotides.

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

- Another extreme: as many duplications as makes sense.
- Resembles genomes when considering them as a sequence of nucleotides.
- A tool for deriving lower bounds.

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

- Another extreme: as many duplications as makes sense.
- Resembles genomes when considering them as a sequence of nucleotides.
- A tool for deriving lower bounds.
- A helping method for sorting permutations.

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

- Another extreme: as many duplications as makes sense.
- Resembles genomes when considering them as a sequence of nucleotides.
- A tool for deriving lower bounds.
- A helping method for sorting permutations.
- **Interesting on its own.**

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

- Another extreme: as many duplications as makes sense.
- Resembles genomes when considering them as a sequence of nucleotides.
- A tool for deriving lower bounds.
- A helping method for sorting permutations.
- Interesting on its own.
- **Example:** $111010010 \rightarrow 000011111$

Background

The Reversal Problem

Sorting by Reversals

Applications

0/1 Sorting

Aspects of the Problem

Results

Sorting 0/1 Sequences

Aspects of the Problem

Background

Aspects of the Problem

Circularity vs. Linearity

Signs

Costs

Results

Sorting 0/1 Sequences

Circularity vs. Linearity

- Genomes of prokaryotes and of organelles are circular.
- Most other genomes are linear.

Background

Aspects of the Problem

Circularity vs. Linearity

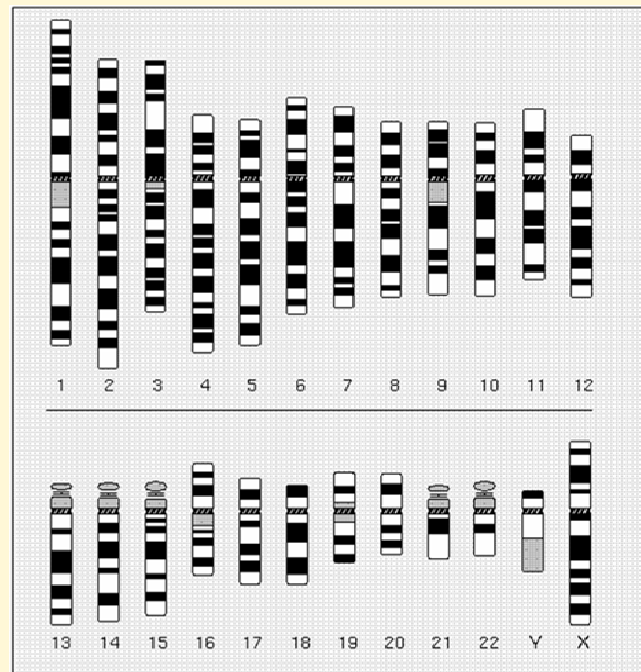
Signs

Costs

Results

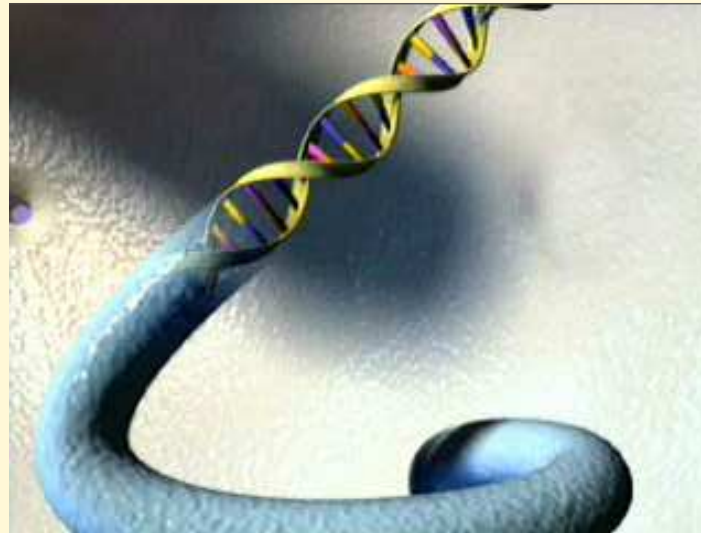
Sorting 0/1 Sequences

Figure 2: Homo sapiens chromosomes.



- DNA has 2 strands:

Figure 3: Visualization of a chromosome.



[Background](#)

[Aspects of the Problem](#)

[Circularity vs. Linearity](#)

[Signs](#)

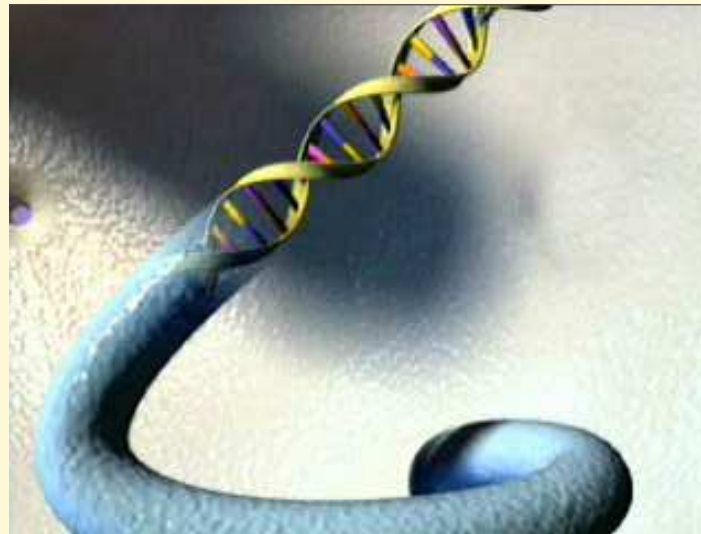
[Costs](#)

[Results](#)

[Sorting 0/1 Sequences](#)

- DNA has 2 strands: **model by signs.**

Figure 3: Visualization of a chromosome.



[Background](#)

[Aspects of the Problem](#)

[Circularity vs. Linearity](#)

[Signs](#)

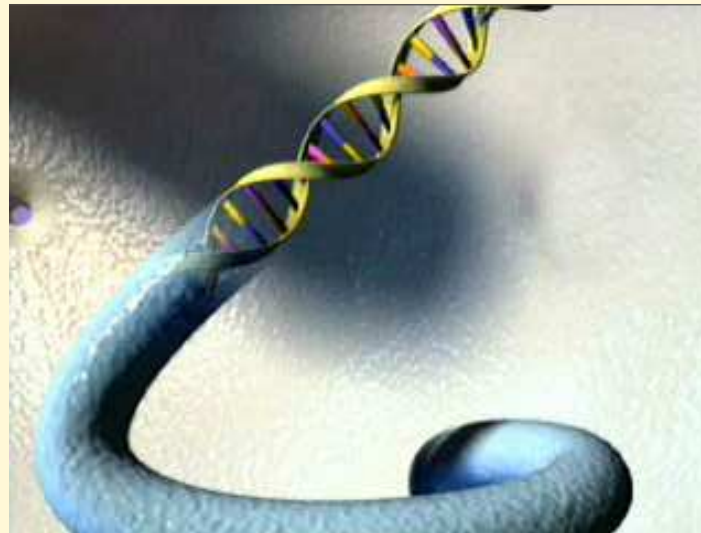
[Costs](#)

[Results](#)

[Sorting 0/1 Sequences](#)

- DNA has 2 strands: **model by signs.**

Figure 3: Visualization of a chromosome.



strand 1 : +1 +4

strand 2 : ... -2 -3 ...

[Background](#)

[Aspects of the Problem](#)

[Circularity vs. Linearity](#)

[Signs](#)

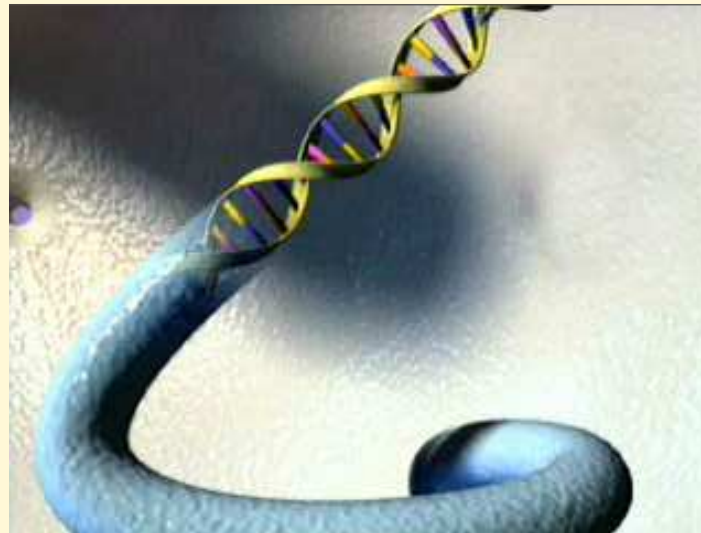
[Costs](#)

[Results](#)

[Sorting 0/1 Sequences](#)

- DNA has 2 strands: **model by signs.**

Figure 3: Visualization of a chromosome.



- Reversals' effect:

strand 1 : +1 +4

strand 2 : ... -2 -3 ...

Background

Aspects of the Problem

Circularity vs. Linearity

Signs

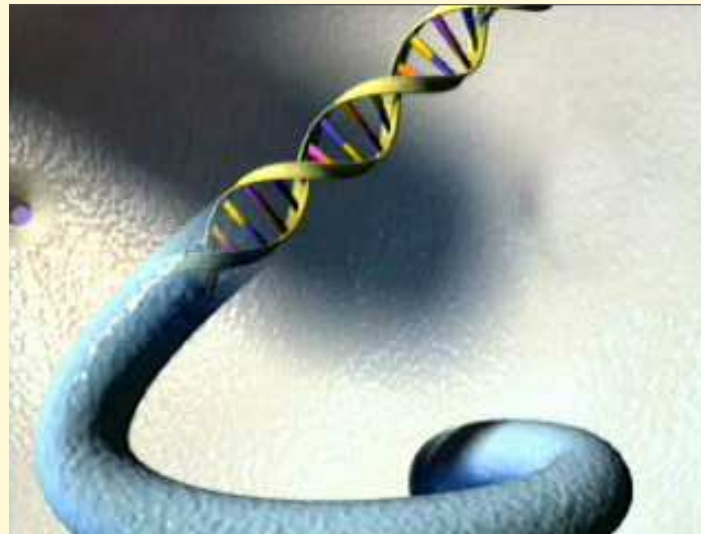
Costs

Results

Sorting 0/1 Sequences

- DNA has 2 strands: **model by signs.**

Figure 3: Visualization of a chromosome.



- Reversals' effect:

strand 1 : +1 +3 +2 +4

strand 2 :

Background

Aspects of the Problem

Circularity vs. Linearity

Signs

Costs

Results

Sorting 0/1 Sequences

- Cost of a reversal sequence equals sums of costs.

[Background](#)

[Aspects of the Problem](#)

[Circularity vs. Linearity](#)

[Signs](#)

[Costs](#)

[Results](#)

[Sorting 0/1 Sequences](#)

- Cost of a reversal sequence equals sums of costs.
- The length-weighted model: reversal of length l has cost $f(l) = l^\alpha$, for $\alpha \geq 0$.

[Background](#)

[Aspects of the Problem](#)

[Circularity vs. Linearity](#)

[Signs](#)

[Costs](#)

[Results](#)

[Sorting 0/1 Sequences](#)

- Cost of a reversal sequence equals sums of costs.
- The length-weighted model: reversal of length l has cost $f(l) = l^\alpha$, for $\alpha \geq 0$.
- The traditional model (unit-cost): $\alpha = 0$.

Background

Aspects of the Problem

Circularity vs. Linearity

Signs

Costs

Results

Sorting 0/1 Sequences

Results

[Background](#)

[Aspects of the Problem](#)

Results

[Previous Results](#)

[This Work: Signs and Circularity](#)

[Sorting 0/1 Sequences](#)

Previous Results

- First results by [Pinter-Skienna'02] for linear cost functions.
- Improved and extended to a wider range of cost functions by [Bender et. al.'04].
- All the above deals with the **unsigned linear** case.

[Background](#)

[Aspects of the Problem](#)

[Results](#)

[Previous Results](#)

[This Work: Signs and Circularity](#)

[Sorting 0/1 Sequences](#)

This Work: Signs and Circularity

Table 1: Approximation ratios for permutations' sorting.

α Value	Unsigned		Signed	
	Linear	Circular	Linear	Circular
$\alpha = 1$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
$1 < \alpha < 2$	$O(\log n)$		$O(\log n)$	

This Work: Signs and Circularity

Table 1: Approximation ratios for permutations' sorting.

α Value	Unsigned		Signed	
	Linear	Circular	Linear	Circular
$\alpha = 1$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
$1 < \alpha < 2$	$O(\log n)$		$O(\log n)$	

Table 2: Approximation ratios for 0/1's sorting.

α Value	Unsigned		Signed	
	Linear	Circular	Linear	Circular
$0 \leq \alpha < 1$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
$\alpha = 1$	1	1	3	3
$1 < \alpha < 2$	$O(1)$	$O(1)$	$O(1)$	$O(1)$

Sorting 0/1 Sequences for $\alpha = 1$

[Background](#)

[Aspects of the Problem](#)

[Results](#)

[Sorting 0/1 Sequences](#)

[Reversals' Properties](#)

[Characterization](#)

[Polynomial-Time Algorithm](#)

[Circular vs. Linear](#)

Reversals' Properties

[Background](#)

[Aspects of the Problem](#)

[Results](#)

[Sorting 0/1 Sequences](#)

[Reversals' Properties](#)

[Characterization](#)

[Polynomial-Time Algorithm](#)

[Circular vs. Linear](#)

Reversals' Properties

- No *Useless*: 0110101

[Background](#)

[Aspects of the Problem](#)

[Results](#)

[Sorting 0/1 Sequences](#)

[Reversals' Properties](#)

[Characterization](#)

[Polynomial-Time Algorithm](#)

[Circular vs. Linear](#)

Reversals' Properties

- No *Useless*: 0110101
- No *Cutting*: 011110010

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Reversals' Properties

- No *Useless*: 0110101
- No *Cutting*: 011110010
- No *Complex*: 01010

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Reversals' Properties

- No *Useless*: 0110101
- No *Cutting*: 011110010
- No *Complex*: 01010
- A non-complex reversal is called *simple*.

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Reversals' Properties

- No *Useless*: 0110101
- No *Cutting*: 011110010
- No *Complex*: 01010
- A non-complex reversal is called *simple*.
- A simple reversal that is not cutting is called *good*.

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Reversals' Properties

- No *Useless*: 0**110101**
- No *Cutting*: 011**110010**
- No *Complex*: 0**1010**
- A non-complex reversal is called *simple*.
- A simple reversal that is not cutting is called *good*.

Theorem 1. *There exists a good optimal sorting reversal sequence.*

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Characterization

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Lemma 2. *When sorting a sequence with good reversals, there exist indices i (0-block) and j (1-block) that take part in no reversals.*

Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

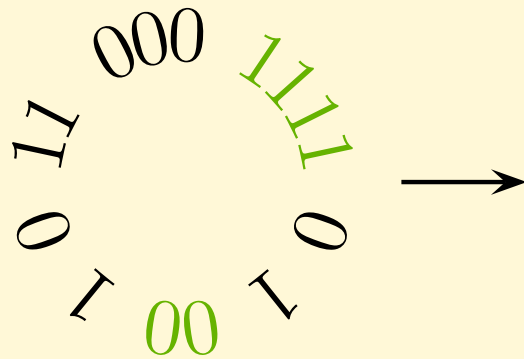
Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Lemma 2. *When sorting a sequence with good reversals, there exist indices i (0-block) and j (1-block) that take part in no reversals.*



Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

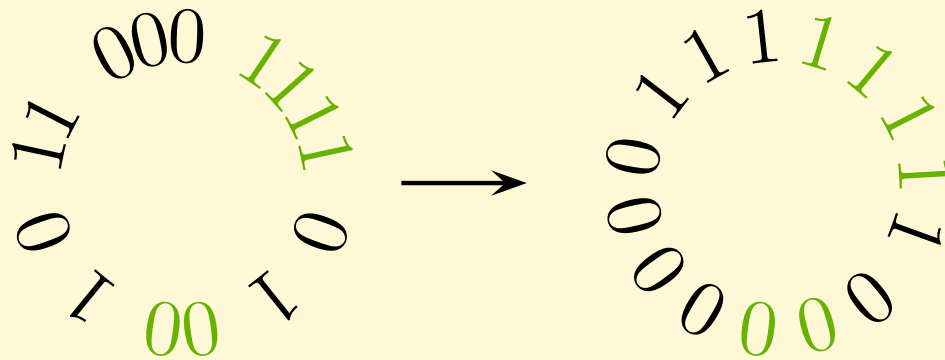
Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Lemma 2. *When sorting a sequence with good reversals, there exist indices i (0-block) and j (1-block) that take part in no reversals.*



Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

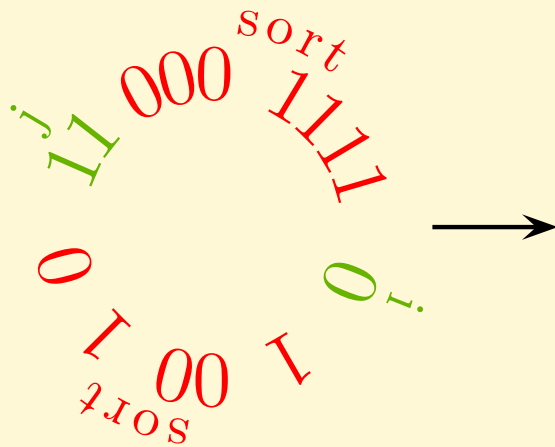
Characterization

Polynomial-Time Algorithm

Circular vs. Linear

circularZeroOneSort

- For all pair of indices i (0-block) and j (1-block):



Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

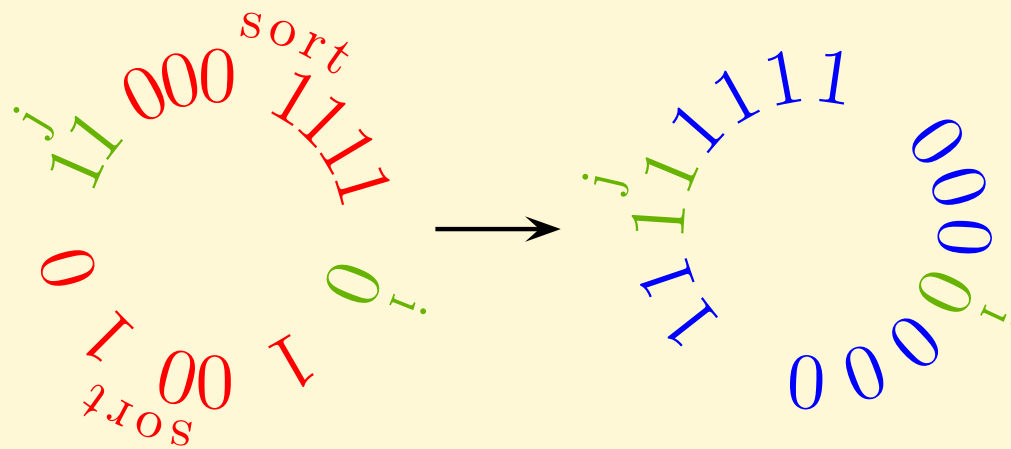
Characterization

Polynomial-Time Algorithm

Circular vs. Linear

circularZeroOneSort

- For all pair of indices i (0-block) and j (1-block):
- Sort the segments (i, j) and (j, i) using `zerOneSort` [Bender et. al.'04].



Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

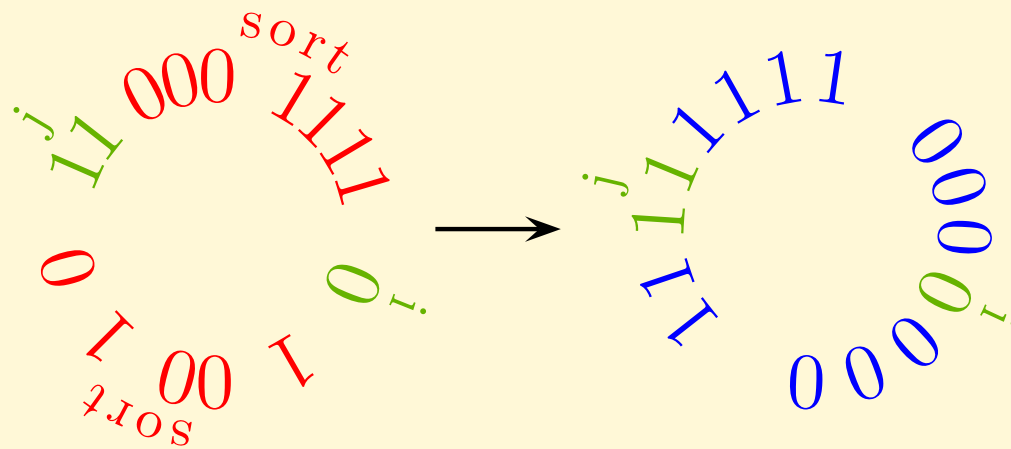
Characterization

Polynomial-Time Algorithm

Circular vs. Linear

circularZeroOneSort

- For all pair of indices i (0-block) and j (1-block):
- Sort the segments (i, j) and (j, i) using `zerOneSort` [Bender et. al.'04].
- Return the minimum over all pairs.



Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear

Can be implemented as follows:

- Time: $O(n^3)$.
- Space: $O(n^2)$.

[Background](#)

[Aspects of the Problem](#)

[Results](#)

[Sorting 0/1 Sequences](#)

[Reversals' Properties](#)

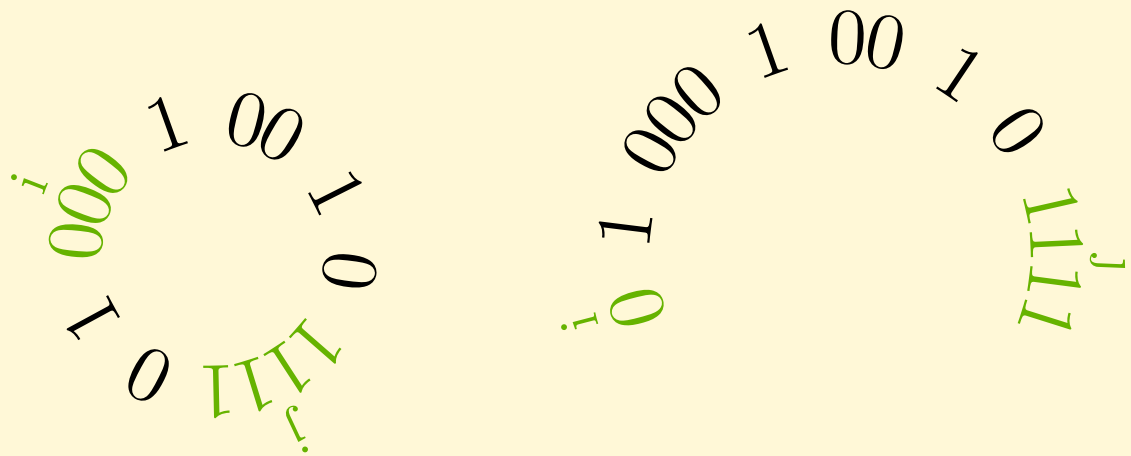
[Characterization](#)

[Polynomial-Time Algorithm](#)

[Circular vs. Linear](#)

Circular vs. Linear

In linear, the indices i and j correspond to the edges of the sequences, whereas in circular the indices i and j are chosen to minimize the sorting cost.



Background

Aspects of the Problem

Results

Sorting 0/1 Sequences

Reversals' Properties

Characterization

Polynomial-Time Algorithm

Circular vs. Linear