On Sorting of Signed Permutations by Prefix and Suffix Reversals and Transpositions

AlCoB'2014, Tarragona, Spain

Carla Negri Lintzmayer
Zanoni Dias

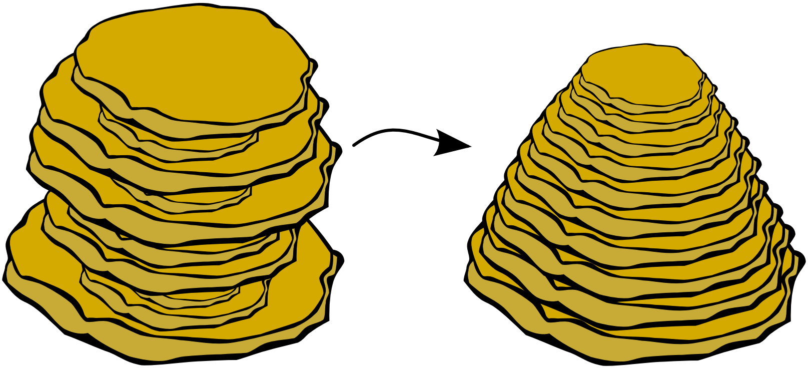
University of Campinas (UNICAMP)
Institute of Computing
Campinas, São Paulo, Brazil

Partially supported by FAPESP (grants 2013/01172-0 and 2013/08293-7), CNPq (grants 477692/2012-5 and 483370/2013-4), and FAEPEX (process 396/2014)

July 3, 2014

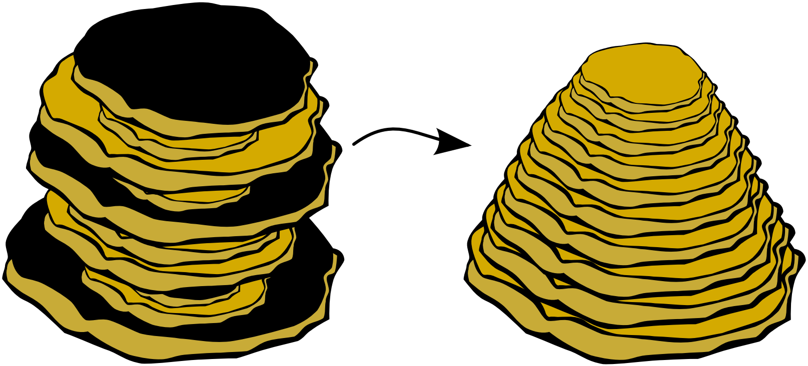
Introduction

Dweighter [1], 1975: *The Pancake Flipping Problem*



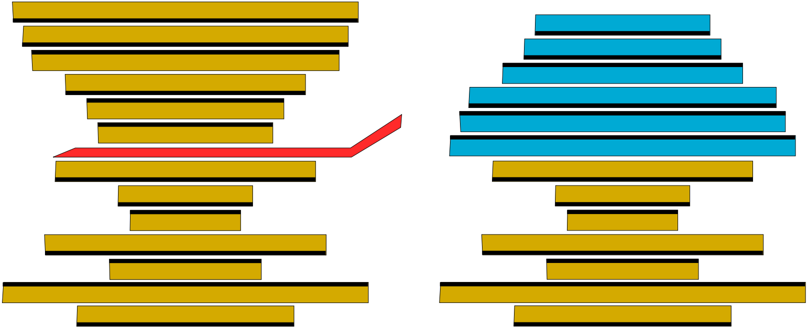
Introduction

Gates and Papadimitriou [2], 1979: *The Burnt Pancake Flipping Problem*



Introduction

Allowed moves: prefix reversals



Introduction

- Reinterpreted as Genome Rearrangement Problems

Introduction

- Reinterpreted as Genome Rearrangement Problems
- Genome Rearrangements: mutations that affect large portions of the genome

Introduction

- Reinterpreted as Genome Rearrangement Problems
- Genome Rearrangements: mutations that affect large portions of the genome
  - Reversals and Transpositions

Introduction

- Reinterpreted as Genome Rearrangement Problems
- Genome Rearrangements: mutations that affect large portions of the genome
  - Reversals and Transpositions
- Mutations allow evolutions and the difference between two genomes indicates their evolutionary distance

Introduction

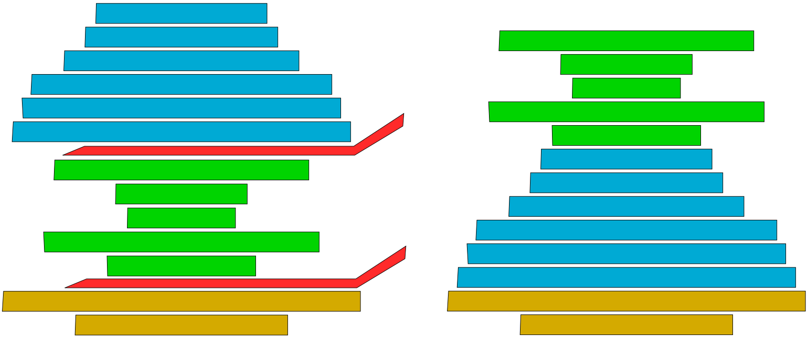
- Reinterpreted as Genome Rearrangement Problems
- Genome Rearrangements: mutations that affect large portions of the genome
  - Reversals and Transpositions
- Mutations allow evolutions and the difference between two genomes indicates their evolutionary distance
- Find scenarios that show how to transform one genome into another

Introduction

- Reinterpreted as Genome Rearrangement Problems
- Genome Rearrangements: mutations that affect large portions of the genome
    - Reversals and Transpositions
- Mutations allow evolutions and the difference between two genomes indicates their evolutionary distance
- Find scenarios that show how to transform one genome into another
    - minimum number of rearrangements that allow the transformation

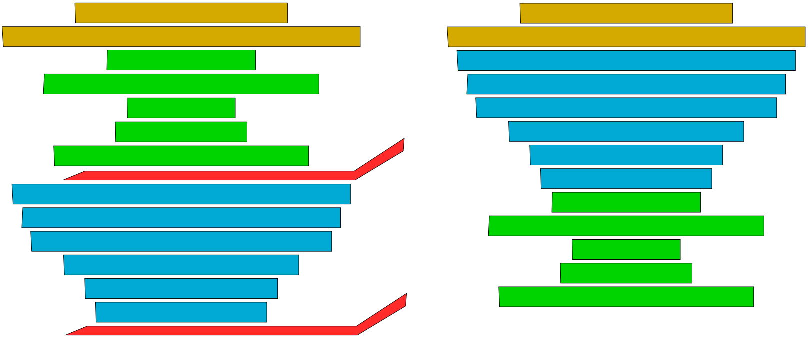
Introduction

Dias and Meidanis [3], 2002: Prefix Transpositions
Sharmin *et al.* [4], 2010: *Pancake Flipping with Two Spatulas*



Introduction

Lintzmayer and Dias [5], 2014: Suffix operations



Definitions

- Permutation:
  $\pi = (\pi_0 = +0 \ \pi_1 \ \pi_2 \ \ldots \ \pi_n \ \pi_{n+1} = +(n+1))$
  where $|\pi_i| \neq |\pi_j|$ for all $i \neq j$

Definitions

- Permutation:
  $\pi = (\pi_0 = +0 \ \pi_1 \ \pi_2 \ \ldots \ \pi_n \ \pi_{n+1} = +(n+1))$
  where $|\pi_i| \neq |\pi_j|$ for all $i \neq j$
- Composition: $\pi \cdot \sigma = (\pi_{\sigma_1} \ \pi_{\sigma_2} \ \ldots \ \pi_{\sigma_n})$

Definitions

- Permutation:
  $\pi = (\pi_0 = +0 \ \pi_1 \ \pi_2 \ \ldots \ \pi_n \ \pi_{n+1} = +(n+1))$
  where $|\pi_i| \neq |\pi_j|$ for all $i \neq j$
- Composition: $\pi \cdot \sigma = (\pi_{\sigma_1} \ \pi_{\sigma_2} \ \ldots \ \pi_{\sigma_n})$
- Identity permutation:
  $\iota_n = (+1 \ +2 \ \ldots \ +(n-1) \ +n)$

Definitions

- Permutation:
  $\pi = (\pi_0 = +0 \ \pi_1 \ \pi_2 \ \ldots \ \pi_n \ \pi_{n+1} = +(n+1))$
  where $|\pi_i| \neq |\pi_j|$ for all $i \neq j$
- Composition: $\pi \cdot \sigma = (\pi_{\sigma_1} \ \pi_{\sigma_2} \ \ldots \ \pi_{\sigma_n})$
- Identity permutation:
  $\iota_n = (+1 \ +2 \ \ldots \ +(n-1) \ +n)$
- Reverse permutation:
  $\eta_n = (-n \ -(n-1) \ \ldots \ -2 \ -1)$

Definitions

- Signed reversal: $\bar{\rho}(i, j)$ with $1 \leq i \leq j \leq n$

$$\begin{array}{rcl} \pi & = & (\pi_1 \ \dots \ \pi_{i-1} \ \underline{\pi_i \ \ \pi_{i+1} \ \dots \ \pi_{j-1} \ \ \pi_j} \ \pi_{j+1} \ \dots \ \pi_n) \\ \pi \cdot \bar{\rho}(i,j) & = & (\pi_1 \ \dots \ \pi_{i-1} \ \underline{-\pi_j \ -\pi_{j-1} \ \dots \ -\pi_{i+1} \ -\pi_i} \ \pi_{j+1} \ \dots \ \pi_n) \end{array}$$

Definitions

- Signed reversal: $\bar{\rho}(i,j)$ with $1 \leq i \leq j \leq n$

$$\begin{array}{rcl} \pi & = & (\pi_1 \ \dots \ \pi_{i-1} \ \underline{\pi_i \ \ \pi_{i+1} \ \dots \ \pi_{j-1} \ \ \pi_j} \ \pi_{j+1} \ \dots \ \pi_n) \\ \pi\cdot\bar{\rho}(i,j) & = & (\pi_1 \ \dots \ \pi_{i-1} \ \underline{-\pi_j \ -\pi_{j-1} \ \dots \ -\pi_{i+1} \ -\pi_i} \ \pi_{j+1} \ \dots \ \pi_n) \end{array}$$

- Example:

$$\begin{array}{rl} \pi = & (-3 \ \underline{+1 \ -5 \ +2 \ +7} \ -4 \ -3) \\ \pi \cdot \bar{\rho}(2,5) = & (-3 \ \underline{-7 \ -2 \ +5 \ -1} \ -4 \ -3) \end{array}$$

Definitions

- Transposition: $\tau(i, j, k)$ with $1 \leq i < j < k \leq n+1$

$$\begin{aligned} \pi &= (\pi_1 \ldots \pi_{i-1}\ \underline{\pi_i\ \ \pi_{i+1} \ldots \pi_{j-1}}\ \underline{\pi_j\ \ \pi_{j+1} \ldots \pi_{k-1}}\ \pi_k \ldots \pi_n) \\ \pi \cdot \tau(i,j,k) &= (\pi_1 \ldots \pi_{i-1}\ \underline{\pi_j\ \ \pi_{j+1} \ldots \pi_{k-1}}\ \underline{\pi_i\ \ \pi_{i+1} \ldots \pi_{j-1}}\ \pi_k \ldots \pi_n) \end{aligned}$$

Definitions

- Transposition: $\tau(i, j, k)$ with $1 \leq i < j < k \leq n+1$

$$
\begin{array}{rcl}
\pi & = & (\pi_1 \ldots \pi_{i-1} \; \underline{\pi_i \;\; \pi_{i+1} \ldots \pi_{j-1}} \; \underline{\pi_j \;\; \pi_{j+1} \ldots \pi_{k-1}} \; \pi_k \ldots \pi_n) \\
\pi \cdot \tau(i,j,k) & = & (\pi_1 \ldots \pi_{i-1} \; \underline{\pi_j \;\; \pi_{j+1} \ldots \pi_{k-1}} \; \underline{\pi_i \;\; \pi_{i+1} \ldots \pi_{j-1}} \; \pi_k \ldots \pi_n)
\end{array}
$$

- Example:

$$
\begin{array}{rl}
\pi = & (-3 \; \underline{+1 \;\; -5} \; \underline{+2 \;\; +7 \;\; -4} \;\; -3) \\
\pi \cdot \tau(2,4,7) = & (-3 \; \underline{+2 \;\; +7 \;\; -4} \; \underline{+1 \;\; -5} \;\; -3)
\end{array}
$$

Definitions

- Prefix reversal (inverts first segment):
  - $\bar{\rho}_p(j) \equiv \bar{\rho}(1, j)$ for $1 \leq j \leq n$

Definitions

- Prefix reversal (inverts first segment):
  - $\bar{\rho}_p(j) \equiv \bar{\rho}(1, j)$ for $1 \leq j \leq n$
- Suffix reversal (inverts last segment):
  - $\bar{\rho}_s(i) \equiv \bar{\rho}(i, n)$ for $1 \leq i \leq n$

Definitions

- Prefix reversal (inverts first segment):
  - $\bar{\rho}_p(j) \equiv \bar{\rho}(1, j)$ for $1 \leq j \leq n$
- Suffix reversal (inverts last segment):
  - $\bar{\rho}_s(i) \equiv \bar{\rho}(i, n)$ for $1 \leq i \leq n$
- Prefix transposition: $\tau_p(j, k) \equiv \tau(1, j, k)$ for $1 < j < k \leq n + 1$

Definitions

- Prefix reversal (inverts first segment):
  - $\bar{\rho}_p(j) \equiv \bar{\rho}(1, j)$ for $1 \leq j \leq n$
- Suffix reversal (inverts last segment):
  - $\bar{\rho}_s(i) \equiv \bar{\rho}(i, n)$ for $1 \leq i \leq n$
- Prefix transposition: $\tau_p(j, k) \equiv \tau(1, j, k)$ for $1 < j < k \leq n + 1$
- Suffix transposition: $\tau_s(i, j) \equiv \tau(i, j, n+1)$ for $1 \leq i < j < n + 1$

Definitions

- Rearrangement model: β

Definitions

- Rearrangement model: β
- Distance: $d_\beta(\pi)$, minimum number of operations in β needed to transform π into ι_n

Definitions

Example for SbSigPR:

$$
\begin{array}{rl}
\pi = & (\underline{+3}\ \ -1\ \ +4\ \ -2) \\
\bar{\rho}_p(1) \rightarrow & (\underline{-3\ \ -1}\ \ +4\ \ -2) \\
\bar{\rho}_p(2) \rightarrow & (\underline{+1\ \ +3\ \ +4}\ \ -2) \\
\bar{\rho}_p(3) \rightarrow & (\underline{-4\ \ -3\ \ -1\ \ -2}) \\
\bar{\rho}_p(4) \rightarrow & (\underline{+2}\ \ +1\ \ +3\ \ +4) \\
\bar{\rho}_p(1) \rightarrow & (\underline{-2\ \ +1}\ \ +3\ \ +4) \\
\bar{\rho}_p(2) \rightarrow & (\underline{-1}\ \ +2\ \ +3\ \ +4) \\
\bar{\rho}_p(1) \rightarrow & (+1\ \ +2\ \ +3\ \ +4)
\end{array}
$$

Definitions

Example for SbSigPR:

$$
\begin{aligned}
\pi &= (\underline{+3}\ \ -1\ \ +4\ \ -2)\\
\bar{\rho}_p(1) &\rightarrow (\underline{-3\ \ -1}\ \ +4\ \ -2)\\
\bar{\rho}_p(2) &\rightarrow (\underline{+1\ \ +3\ \ +4}\ \ -2)\\
\bar{\rho}_p(3) &\rightarrow (\underline{-4\ \ -3\ \ -1\ \ -2})\\
\bar{\rho}_p(4) &\rightarrow (\underline{+2}\ \ +1\ \ +3\ \ +4)\\
\bar{\rho}_p(1) &\rightarrow (\underline{-2\ \ +1}\ \ +3\ \ +4)\\
\bar{\rho}_p(2) &\rightarrow (\underline{-1}\ \ +2\ \ +3\ \ +4)\\
\bar{\rho}_p(1) &\rightarrow (+1\ \ +2\ \ +3\ \ +4)
\end{aligned}
$$

$$
\begin{aligned}
\pi &= (\underline{+3\ \ -1\ \ +4}\ \ -2)\\
\bar{\rho}_p(3) &\rightarrow (\underline{-4\ \ +1\ \ -3\ \ -2})\\
\bar{\rho}_p(4) &\rightarrow (\underline{+2\ \ +3}\ \ -1\ \ +4)\\
\bar{\rho}_p(2) &\rightarrow (\underline{-3\ \ -2\ \ -1}\ \ +4)\\
\bar{\rho}_p(3) &\rightarrow (+1\ \ +2\ \ +3\ \ +4)
\end{aligned}
$$

Definitions

Example for SbSigPR:

$$
\begin{aligned}
\pi &= (\underline{+3} \;\; -1 \;\; +4 \;\; -2) \\
\bar{\rho}_p(1) &\rightarrow (\underline{-3 \;\; -1} \;\; +4 \;\; -2) \\
\bar{\rho}_p(2) &\rightarrow (\underline{+1 \;\; +3 \;\; +4} \;\; -2) \\
\bar{\rho}_p(3) &\rightarrow (\underline{-4 \;\; -3 \;\; -1 \;\; -2}) \\
\bar{\rho}_p(4) &\rightarrow (\underline{+2} \;\; +1 \;\; +3 \;\; +4) \\
\bar{\rho}_p(1) &\rightarrow (\underline{-2 \;\; +1} \;\; +3 \;\; +4) \\
\bar{\rho}_p(2) &\rightarrow (\underline{-1} \;\; +2 \;\; +3 \;\; +4) \\
\bar{\rho}_p(1) &\rightarrow (+1 \;\; +2 \;\; +3 \;\; +4)
\end{aligned}
$$

$$
\begin{aligned}
\pi &= (\underline{+3 \;\; -1 \;\; +4} \;\; -2) \\
\bar{\rho}_p(3) &\rightarrow (\underline{-4 \;\; +1 \;\; -3 \;\; -2}) \\
\bar{\rho}_p(4) &\rightarrow (\underline{+2 \;\; +3} \;\; -1 \;\; +4) \\
\bar{\rho}_p(2) &\rightarrow (\underline{-3 \;\; -2 \;\; -1} \;\; +4) \\
\bar{\rho}_p(3) &\rightarrow (+1 \;\; +2 \;\; +3 \;\; +4)
\end{aligned}
$$

$$d_{\bar{\rho}_p}(\pi) = 4$$

Definitions

- Breakpoint: occurs between two consecutive elements of π that should not be consecutive

Definitions

- Breakpoint: occurs between two consecutive elements of π that should not be consecutive
- Example:

$$(+0 \;.\; -3 \; -2 \,.\, -4 \,.\, -5 \,.\, +1 \;.\; +6)$$

Definitions

- Breakpoint: occurs between two consecutive elements of π that should not be consecutive
- Example:

$$(+0\ \textcolor{red}{\blacksquare}\ -3\ \ -2 \blacksquare -4 \blacksquare -5 \blacksquare +1\ \textcolor{red}{\blacksquare}\ +6)$$

- Strip: maximal subsequence of π without breakpoints

Problems

Problem	Best approximation factor	Complexity
SbR	1.375 [6]	NP-hard [7]
SbSigR	-	P [8]
SbT	1.375 [9]	NP-hard [10]
SbRT	≈2.83 [11]	?
SbSigRT	2^* [12]	?
SbPR	2 [13]	NP-hard [14]
SbSigPR	2 [15]	?
SbPT	2 [3]	?
SbPRPT	3 [4]	?

SbSigPRSigSR

Tries to remove one breakpoint with one operation:

1. $\pi_j = -\pi_1 + 1$, $2 \leq j \leq n$: $\bar{\rho}_p(j-1)$

SbSigPRSigSR

Tries to remove one breakpoint with one operation:

1. $\pi_j = -\pi_1 + 1,\ 2 \leq j \leq n$: $\bar{\rho}_p(j-1)$
2. $\pi_i = -\pi_n - 1,\ 1 \leq i \leq n-1$: $\bar{\rho}_s(i+1)$

SbSigPRSigSR

Tries to remove one breakpoint with one operation:

1. $\pi_j = -\pi_1 + 1$, $2 \le j \le n$: $\bar{\rho}_p(j-1)$
2. $\pi_i = -\pi_n - 1$, $1 \le i \le n-1$: $\bar{\rho}_s(i+1)$

SbSigPRSigSR

Tries to remove one breakpoint with one operation:

1. $\pi_j = -\pi_1 + 1$, $2 \leq j \leq n$: $\bar{\rho}_p(j-1)$
2. $\pi_i = -\pi_n - 1$, $1 \leq i \leq n-1$: $\bar{\rho}_s(i+1)$

Tries to remove one breakpoint with two operations:

1. $\pi_j = -\pi_i - 1$, $1 \leq i < j \leq n$: $\bar{\rho}_p(j) \cdot \bar{\rho}_p(j-i)$

SbSigPRSigSR

Tries to remove one breakpoint with one operation:

1. $\pi_j = -\pi_1 + 1$, $2 \leq j \leq n$: $\bar{\rho}_p(j-1)$
2. $\pi_i = -\pi_n - 1$, $1 \leq i \leq n-1$: $\bar{\rho}_s(i+1)$

Tries to remove one breakpoint with two operations:

1. $\pi_j = -\pi_i - 1$, $1 \leq i < j \leq n$: $\bar{\rho}_p(j) \cdot \bar{\rho}_p(j-i)$
2. $\pi_j = \pi_i + 1$, $0 \leq i+1 < j \leq n$: $\bar{\rho}_p(i) \cdot \bar{\rho}_p(j-1)$

SbSigPRSigSR

Tries to remove one breakpoint with one operation:

1. $\pi_j = -\pi_1 + 1$, $2 \leq j \leq n$: $\bar{\rho}_p(j-1)$
2. $\pi_i = -\pi_n - 1$, $1 \leq i \leq n-1$: $\bar{\rho}_s(i+1)$

Tries to remove one breakpoint with two operations:

1. $\pi_j = -\pi_i - 1$, $1 \leq i < j \leq n$: $\bar{\rho}_p(j) \cdot \bar{\rho}_p(j-i)$
2. $\pi_j = \pi_i + 1$, $0 \leq i+1 < j \leq n$: $\bar{\rho}_p(i) \cdot \bar{\rho}_p(j-1)$
3. $\pi_i = -\pi_j + 1$, $1 \leq i < j \leq n$:
   $\bar{\rho}_s(i) \cdot \bar{\rho}_s(n+1-(j-i))$

SbSigPRSigSR

Otherwise, π is of the three forms:

1. η_n

SbSigPRSigSR

Otherwise, π is of the three forms:

1. η_n
   - one signed prefix/suffix reversal sorts it

SbSigPRSigSR

Otherwise, π is of the three forms:

1. η_n
   - one signed prefix/suffix reversal sorts it
2. $\sigma^1 = (\underbrace{p_b{+}1 \ \ldots \ n}_{\ell_b+1} \ \underbrace{p_{b-1}{+}1 \ \ldots \ p_b}_{\ell_b} \ \ldots\ldots \ \underbrace{1 \ \ldots \ p_1}_{\ell_1})$

SbSigPRSigSR

Otherwise, π is of the three forms:

1. η_n
   - one signed prefix/suffix reversal sorts it
2. $\sigma^1 = (\underbrace{p_b{+}1 \ \dots \ n}_{\ell_b+1} \ \underbrace{p_{b-1}{+}1 \ \dots \ p_b}_{\ell_b} \ \dots\dots \ \underbrace{1 \ \dots \ p_1}_{\ell_1})$
   - at most $b + 2$ reversals sort it

SbSigPRSigSR

Otherwise, π is of the three forms:

1. η_n
   - one signed prefix/suffix reversal sorts it
2. $\sigma^1 = (\underbrace{p_b{+}1 \ \dots \ n}_{\ell_b+1} \ \underbrace{p_{b-1}{+}1 \ \dots \ p_b}_{\ell_b} \ \dots\dots \ \underbrace{1 \ \dots \ p_1}_{\ell_1})$
   - at most $b + 2$ reversals sort it
3. $\sigma^2 = (\underbrace{-p_1 \ \dots \ -1}_{\ell_1} \ \underbrace{-p_2 \ \dots \ -(p_1{+}1)}_{\ell_2} \ \dots\dots$
   $\underbrace{-n \ \dots \ -(p_b{+}1)}_{\ell_{b+1}})$

SbSigPRSigSR

Otherwise, π is of the three forms:

1. η_n
   - one signed prefix/suffix reversal sorts it
2. $\sigma^1 = (\underbrace{p_b{+}1 \;\ldots\; n}_{\ell_b+1}\ \underbrace{p_{b-1}{+}1 \;\ldots\; p_b}_{\ell_b}\ \ldots\ldots\ \underbrace{1 \;\ldots\; p_1}_{\ell_1})$
   - at most $b+2$ reversals sort it
3. $\sigma^2 = (\underbrace{-p_1 \;\ldots\; -1}_{\ell_1}\ \underbrace{-p_2 \;\ldots\; -(p_1{+}1)}_{\ell_2}\ \ldots\ldots\ \underbrace{-n \;\ldots\; -(p_b{+}1)}_{\ell_{b+1}})$
   - at most $b+2$ reversals sort it

SbSigPRSigSR

$$d_{\bar{\rho}_p\bar{\rho}_s}(\pi) \leq 2b_{\bar{\rho}_p\bar{\rho}_s}(\pi) + 1 \tag{1}$$

SbSigPRSigSR

$$d_{\bar{\rho}_p \bar{\rho}_s}(\pi) \leq 2b_{\bar{\rho}_p \bar{\rho}_s}(\pi) + 1 \quad (1)$$

$$d_{\bar{\rho}_p \bar{\rho}_s}(\pi) \geq b_{\bar{\rho}_p \bar{\rho}_s}(\pi) \quad (2)$$

SbSigPRSigSR

$$d_{\bar{\rho_p}\bar{\rho_s}}(\pi) \leq 2b_{\bar{\rho_p}\bar{\rho_s}}(\pi) + 1 \qquad (1)$$

$$d_{\bar{\rho_p}\bar{\rho_s}}(\pi) \geq b_{\bar{\rho_p}\bar{\rho_s}}(\pi) \qquad (2)$$

$$\lim_{b(\pi)\to\infty} \frac{2b(\pi)+1}{b(\pi)} = 2 + \lim_{b(\pi)\to\infty} \frac{1}{b(\pi)} = 2 + \epsilon \qquad (3)$$

SbSigPRPT

If $\pi_1 \neq 1$, tries to remove two breakpoints with one operation:

1. Since $\pi \cdot \tau_p(i,j) = (\underline{\pi_i \ \ldots \ \pi_{j-1}} \ \underline{\pi_1 \ \ldots \ \pi_{i-1}} \ \ldots \ \pi_n)$, we must find $\pi_{j-1} = \pi_1 - 1$ and $\pi_{i-1} = \pi_j - 1$

SbSigPRPT

If $\pi_1 \neq 1$, tries to remove two breakpoints with one operation:

1. Since $\pi \cdot \tau_p(i,j) = (\underline{\pi_i \ \ldots \ \pi_{j-1}} \ \underline{\pi_1 \ \ldots \ \pi_{i-1}} \ \ldots \ \pi_n)$, we must find $\pi_{j-1} = \pi_1 - 1$ and $\pi_{i-1} = \pi_j - 1$
   - To maintain our approximation, $\pi_i \neq 1$

SbSigPRPT

If $\pi_1 \neq 1$, tries to remove one breakpoint with one operation by increasing the first strip:

1. Let $\pi = (k{+}1\ k{+}2\ \ldots\ k{+}(i-1)\ \pi_i\ \ldots\ldots)$

SbSigPRPT

If $\pi_1 \neq 1$, tries to remove one breakpoint with one operation by increasing the first strip:

1. Let $\pi = (k{+}1 \ k{+}2 \ \ldots \ k{+}(i-1) \ \pi_i \ \ldots\ldots)$
2. If $\pi_j = k + i = \pi_{i-1} + 1$ exists, then $\tau_p(i, j)$

SbSigPRPT

If $\pi_1 \neq 1$, tries to remove one breakpoint with one operation by increasing the first strip:

1. Let $\pi = (k{+}1\ k{+}2\ \ \ldots\ \ k{+}(i-1)\ \pi_i\ \ \ldots\ldots)$
2. If $\pi_j = k + i = \pi_{i-1} + 1$ exists, then $\tau_p(i, j)$
3. If $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i, j)$

SbSigPRPT

If $\pi_1 \neq 1$, tries to remove one breakpoint with one operation by increasing the first strip:

1. Let $\pi = (k{+}1\ k{+}2\ \ldots\ k{+}(i-1)\ \pi_i\ \ldots\ldots)$
2. If $\pi_j = k + i = \pi_{i-1} + 1$ exists, then $\tau_p(i, j)$
3. If $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i, j)$
4. If $\pi_{j+1} = -\pi_1 + 1$ exists, then $\bar{\rho}_p(j)$, for $1 \leq j \leq n$

SbSigPRPT

If $\pi_1 = 1$, send the first strip to the end of the permutation:

1. It will be removed only when n is sent there

SbSigPRPT

If $\pi_1 = 1$, send the first strip to the end of the permutation:

1. It will be removed only when n is sent there
2. Which guarantees that $\pi_1 = 1$ again at most one more time

SbSigPRPT

If $\pi_1 = 1$, send the first strip to the end of the permutation:

1. It will be removed only when n is sent there
2. Which guarantees that $\pi_1 = 1$ again at most one more time
3. Therefore, it will be possible to remove at least one breakpoint until the end of the sorting

SbSigPRPT

If $\pi_1 = 1$, send the first strip to the end of the permutation:

1. It will be removed only when n is sent there
2. Which guarantees that $\pi_1 = 1$ again at most one more time
3. Therefore, it will be possible to remove at least one breakpoint until the end of the sorting
   - using at most two extra operations that do not remove breakpoints

SbSigPRPT

$$d_{\bar{\rho}_p \tau_p}(\pi) \leq b_{\bar{\rho}_p}(\pi) + 2 \quad (4)$$

SbSigPRPT

$$d_{\bar{\rho}_p \tau_p}(\pi) \leq b_{\bar{\rho}_p}(\pi) + 2 \quad (4)$$

$$d_{\bar{\rho}_p \tau_p}(\pi) \geq \frac{b_{\bar{\rho}_p}(\pi)}{2} \quad (5)$$

SbSigPRPT

$$d_{\bar{\rho}_p \tau_p}(\pi) \leq b_{\bar{\rho}_p}(\pi) + 2 \tag{4}$$

$$d_{\bar{\rho}_p \tau_p}(\pi) \geq \frac{b_{\bar{\rho}_p}(\pi)}{2} \tag{5}$$

$$\lim_{b(\pi)\to\infty} \frac{b(\pi) + 2}{\frac{b(\pi)}{2}} = 2 + \lim_{b(\pi)\to\infty} \frac{4}{b(\pi)} = 2 + \epsilon \tag{6}$$

SbSigPRPTSigSRST

Tries to remove two breakpoints with one operation:

1. if $\pi_{j-1} = \pi_1 - 1$ and $\pi_{i-1} = \pi_j - 1$ exists, $2 \leq i < j \leq n$, then $\tau_p(i, j)$

SbSigPRPTSigSRST

Tries to remove two breakpoints with one operation:

1. if $\pi_{j-1} = \pi_1 - 1$ and $\pi_{i-1} = \pi_j - 1$ exists, $2 \leq i < j \leq n$, then $\tau_p(i,j)$
2. if $\pi_i = \pi_n + 1$ and $\pi_j = \pi_{i-1} + 1$ exists, $2 \leq i < j \leq n$, then $\tau_s(i,j)$

SbSigPRPTSigSRST

Tries to remove two breakpoints with one operation:

1. if $\pi_{j-1} = \pi_1 - 1$ and $\pi_{i-1} = \pi_j - 1$ exists, $2 \leq i < j \leq n$, then $\tau_p(i, j)$
2. if $\pi_i = \pi_n + 1$ and $\pi_j = \pi_{i-1} + 1$ exists, $2 \leq i < j \leq n$, then $\tau_s(i, j)$
3. neither $\pi_{i-1} = n$ and $\pi_i = 1$ nor $\pi_{j-1} = -1$ and $\pi_j = -n$ can happen

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ \ldots\ k+(i-1)\ \pi_i\ \ \ldots\ldots)$, to increase the first strip with a prefix transposition:
   1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
    1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
    2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
   1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
   2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$
2. let $\pi = (\ldots\ldots\ \pi_{j-1}\ k+1\ k+2\ \ldots\ k+x)$, to increase the last strip with a suffix transposition:

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
   1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
   2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$
2. let $\pi = (\ldots\ldots\ \pi_{j-1}\ k+1\ k+2\ \ldots\ k+x)$, to increase the last strip with a suffix transposition:
   1. if $\pi_{i-1} = \pi_j - 1 = k$, $i \geq 2$, exists, then $\tau_s(i,j)$

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
   1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
   2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$
2. let $\pi = (\ldots\ldots\ \pi_{j-1}\ k+1\ k+2\ \ldots\ k+x)$, to increase the last strip with a suffix transposition:
   1. if $\pi_{i-1} = \pi_j - 1 = k$, $i \geq 2$, exists, then $\tau_s(i,j)$
   2. if $\pi_i = \pi_n + 1 = k+x+1$ exists, then $\tau_s(i,j)$

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
   1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
   2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$
2. let $\pi = (\ldots\ldots\ \pi_{j-1}\ k+1\ k+2\ \ldots\ k+x)$, to increase the last strip with a suffix transposition:
   1. if $\pi_{i-1} = \pi_j - 1 = k$, $i \geq 2$, exists, then $\tau_s(i,j)$
   2. if $\pi_i = \pi_n + 1 = k+x+1$ exists, then $\tau_s(i,j)$
3. again, we cannot separate n and 1 or -1 and $-n$

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
   1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
   2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$
2. let $\pi = (\ldots\ldots\ \pi_{j-1}\ k+1\ k+2\ \ldots\ k+x)$, to increase the last strip with a suffix transposition:
   1. if $\pi_{i-1} = \pi_j - 1 = k$, $i \geq 2$, exists, then $\tau_s(i,j)$
   2. if $\pi_i = \pi_n + 1 = k+x+1$ exists, then $\tau_s(i,j)$
3. again, we cannot separate n and 1 or -1 and $-n$
4. if $\pi_{j+1} = -\pi_1 + 1$, $1 \leq j \leq n-1$, exists, then $\bar{\rho}_p(j)$

SbSigPRPTSigSRST

Tries to remove one breakpoint with one operation:

1. let $\pi = (k+1\ k+2\ \ldots\ k+(i-1)\ \pi_i\ \ldots\ldots)$, to increase the first strip with a prefix transposition:
    1. if $\pi_j = k+i = \pi_{i-1}+1$, $j \leq n$, exists, then $\tau_p(i,j)$
    2. if $\pi_{j-1} = k = \pi_1 - 1$ exists, then $\tau_p(i,j)$
2. let $\pi = (\ldots\ldots\ \pi_{j-1}\ k+1\ k+2\ \ldots\ k+x)$, to increase the last strip with a suffix transposition:
    1. if $\pi_{i-1} = \pi_j - 1 = k$, $i \geq 2$, exists, then $\tau_s(i,j)$
    2. if $\pi_i = \pi_n + 1 = k+x+1$ exists, then $\tau_s(i,j)$
3. again, we cannot separate n and 1 or -1 and $-n$
4. if $\pi_{j+1} = -\pi_1 + 1$, $1 \leq j \leq n-1$, exists, then $\bar{\rho}_p(j)$
5. if $\pi_{i-1} = -\pi_n - 1$, $2 \leq i \leq n$, exists, then $\bar{\rho}_s(i)$

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)
2. $(\underline{1\ 2\ \ldots\ k}\ \ldots\ -(k+1)\ \ldots\ -(i-1)\ \ldots\ \underline{i\ i+1\ \ldots\ n})$

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)
2. $(\underline{1\ 2\ \ldots\ k}\ \ldots\ -(k+1)\ \ldots\ -(i-1)\ \ldots\ \underline{i\ i+1\ \ldots\ n})$
3. $(\underline{-n\ \ -(n-1)\ldots-i}\ \ldots\ (i-1)\ \ldots\ (k+1)\ \ldots\ \underline{-k\ \ -(k-1)\ldots-1})$

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)
2. $(\underline{1\ 2\ \ldots\ k}\ \ldots\ -(k+1)\ \ldots\ -(i-1)\ \ldots\ \underline{i\ i+1\ \ldots\ n})$
3. $(\underline{-n\ -(n-1)\ldots -i}\ \ldots\ (i-1)\ \ldots\ (k+1)\ \ldots\ \underline{-k\ -(k-1)\ldots -1})$
4. $(\underline{k+1\ k+2\ \ldots\ n}\ \underline{1\ 2\ \ldots\ k})$

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)
2. $(\underline{1\ 2\ \ldots\ k}\ \ldots\ -(k+1)\ \ldots\ -(i-1)\ \ldots\ \underline{i\ i+1\ \ldots\ n})$
3. $(\underline{-n\ \ -(n-1)\ldots-i}\ \ldots\ (i-1)\ \ldots\ (k+1)\ \ldots\ \underline{-k\ \ -(k-1)\ldots-1})$
4. $(\underline{k+1\ k+2\ \ldots\ n}\ \underline{1\ 2\ \ldots\ k})$
5. $(\underline{-k\ \ -(k-1)\ \ldots\ -1}\ \underline{-n\ \ -(n-1)\ \ldots\ -(k+1)})$

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)
2. $(\underline{1\ 2\ \ldots\ k}\ \ldots\ -(k+1)\ \ldots\ -(i-1)\ \ldots\ \underline{i\ i+1\ \ldots\ n})$
3. $(\underline{-n\ -(n-1)\ldots -i}\ \ldots\ (i-1)\ \ldots\ (k+1)\ \ldots\ \underline{-k\ -(k-1)\ldots -1})$
4. $(\underline{k+1\ k+2\ \ldots\ n}\ \underline{1\ 2\ \ldots\ k})$
5. $(\underline{-k\ -(k-1)\ \ldots\ -1}\ \underline{-n\ -(n-1)\ \ldots\ -(k+1)})$

SbSigPRPTSigSRST

Otherwise, π is of one of the five forms:

1. η_n (one reversal sorts it)
2. $(\underline{1\ 2\ \ldots\ k}\ \ldots\ -(k+1)\ \ldots\ -(i-1)\ \ldots\ \underline{i\ i+1\ \ldots\ n})$
3. $(\underline{-n\ \ -(n-1)\ldots -i}\ \ldots\ (i-1)\ \ldots\ (k+1)\ \ldots\ \underline{-k\ \ -(k-1)\ldots -1})$
4. $(\underline{k+1\ k+2\ \ldots\ n}\ \underline{1\ 2\ \ldots\ k})$
5. $(\underline{-k\ \ -(k-1)\ \ \ldots\ \ -1}\ \underline{-n\ \ -(n-1)\ \ \ldots\ \ -(k+1)})$

For the last four types, we must apply a prefix transposition to concatenate the first strip with the last one

SbSigPRPTSigSRST

$$d_{\bar{\rho}_p \tau_p \bar{\rho}_s \tau_s}(\pi) \leq b_{\bar{\rho}_p \bar{\rho}_s}(\pi) + 2 \tag{7}$$

SbSigPRPTSigSRST

$$d_{\bar{\rho}_p \tau_p \bar{\rho}_s \tau_s}(\pi) \leq b_{\bar{\rho}_p \bar{\rho}_s}(\pi) + 2 \tag{7}$$

$$d_{\bar{\rho}_p \tau_p \bar{\rho}_s \tau_s}(\pi) \geq \frac{b_{\bar{\rho}_p \bar{\rho}_s}(\pi)}{2} \tag{8}$$

SbSigPRPTSigSRST

$$d_{\bar{\rho}_p \tau_p \bar{\rho}_s \tau_s}(\pi) \leq b_{\bar{\rho}_p \bar{\rho}_s}(\pi) + 2 \tag{7}$$

$$d_{\bar{\rho}_p \tau_p \bar{\rho}_s \tau_s}(\pi) \geq \frac{b_{\bar{\rho}_p \bar{\rho}_s}(\pi)}{2} \tag{8}$$

$$\lim_{b(\pi) \to \infty} \frac{b(\pi) + 2}{\frac{b(\pi)}{2}} = 2 + \lim_{b(\pi) \to \infty} \frac{4}{b(\pi)} = 2 + \epsilon \tag{9}$$

Results

- All algorithms are $O(n^2)$

Results

- All algorithms are $O(n^2)$
- They ran under a set of arbitrary permutations: 10000 of each size n, with n varying between 10 and 1000 in intervals of 5

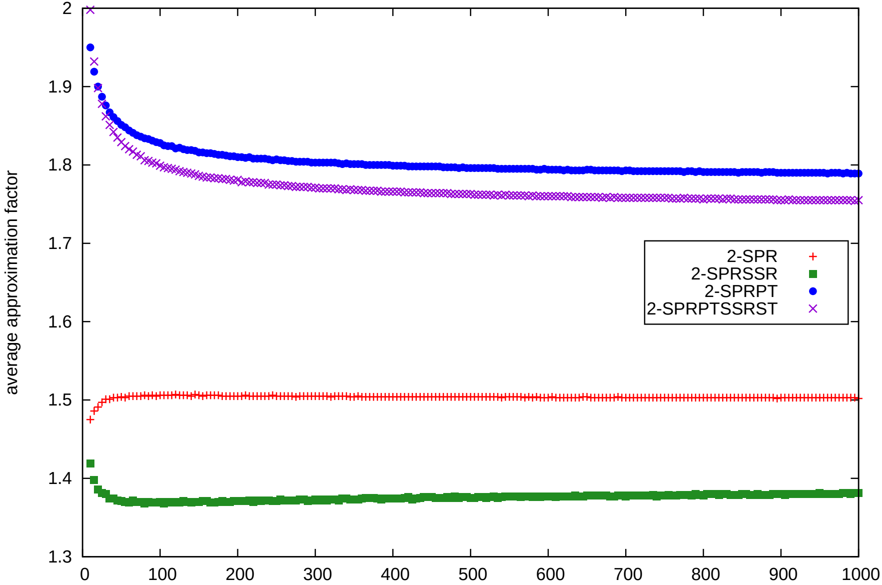
Results

- All algorithms are $O(n^2)$
- They ran under a set of arbitrary permutations: 10000 of each size n, with n varying between 10 and 1000 in intervals of 5
- Approximation factors of the graph are an average between the 10000 of each size n

Results

- All algorithms are $O(n^2)$
- They ran under a set of arbitrary permutations: 10000 of each size n, with n varying between 10 and 1000 in intervals of 5
- Approximation factors of the graph are an average between the 10000 of each size n
- Each approximation factor was calculated using the theoretical lower bound

Results



average approximation factor
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
0
100
200
300
400
500
600
700
800
900
1000
2-SPR
2-SPRSSR
2-SPRPT
2-SPRPTSSRST

Results

Since all algorithms are asymptotic, it is expected to have approximation factors above 2:

- $\textsc{SbSigPRSigSR}$: only when $n = 10$, for one permutation

Results

Since all algorithms are asymptotic, it is expected to have approximation factors above 2:

- $\textsc{SbSigPRSigSR}$: only when $n = 10$, for one permutation
- $\textsc{SbSigPRPT}$: only when $n \leq 100$, for 0.41% of the permutations (72.13% when $n \leq 20$)

Results

Since all algorithms are asymptotic, it is expected to have approximation factors above 2:

- $\textsc{SbSigPRSigSR}$: only when $n = 10$, for one permutation
- $\textsc{SbSigPRPT}$: only when $n \leq 100$, for 0.41% of the permutations (72.13% when $n \leq 20$)
- $\textsc{SbSigPRPTSigSRST}$: only when $n \leq 105$, for 0.44% of the permutations (76.62% when $n \leq 20$)

Conclusions

- We presented the first results for three sorting problems involving signed prefix and suffix operations: SbSigPRSigSR, SbSigPRPT and SbSigPRPTSigSRST

Conclusions

- We presented the first results for three sorting problems involving signed prefix and suffix operations: SbSigPRSigSR, SbSigPRPT and SbSigPRPTSigSRST
- All algorithms established a good approximation factor for these problems

Conclusions

- We presented the first results for three sorting problems involving signed prefix and suffix operations: SbSigPRSigSR, SbSigPRPT and SbSigPRPTSigSRST
- All algorithms established a good approximation factor for these problems
- We are currently working on these problems and their unsigned versions

On Sorting of Signed Permutations by Prefix and Suffix Reversals and Transpositions

AlCoB'2014, Tarragona, Spain

Thank you!

July 3, 2014

References I

[1] H. Dweighter, "Problem E2569," *American Mathematical Monthly*, vol. 82, p. 1010, 1975.

[2] W. H. Gates and C. H. Papadimitriou, "Bounds for Sorting by Prefix Reversal," *Discrete Mathematics*, vol. 27, no. 1, pp. 47–57, 1979.

[3] Z. Dias and J. Meidanis, "Sorting by Prefix Transpositions," in *Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE'2002)*, (London, UK), pp. 65–76, Springer-Verlag, 2002.

References II

[4] M. Sharmin, R. Yeasmin, M. Hasan, A. Rahman, and M. S. Rahman, "Pancake Flipping with Two Spatulas," *Electronic Notes in Discrete Mathematics*, vol. 36, pp. 231–238, 2010.

[5] C. N. Lintzmayer and Z. Dias, "Sorting permutations by prefix and suffix versions of reversals and transpositions," in *LATIN 2014: Theoretical Informatics* (A. Pardo and A. Viola, eds.), vol. 8392 of *Lecture Notes in Computer Science*, pp. 671–682, Springer, 2014.

References III

[6] P. Berman, S. Hannenhalli, and M. Karpinski, "1.375-Approximation Algorithm for Sorting by Reversals," in *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'2002)*, (London, UK), pp. 200–210, Springer-Verlag, 2002.

[7] A. Caprara, "Sorting Permutations by Reversals and Eulerian Cycle Decompositions," *SIAM Journal on Discrete Mathematics*, vol. 12, no. 1, pp. 91–110, 1999.

References IV

[8] S. Hannenhalli and P. A. Pevzner, "Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals," *Journal of the ACM*, vol. 46, no. 1, pp. 1–27, 1999.

[9] I. Elias and T. Hartman, "A 1.375-Approximation Algorithm for Sorting by Transpositions," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 3, no. 4, pp. 369–379, 2006.

[10] L. Bulteau, G. Fertin, and I. Rusu, "Sorting by Transpositions is Difficult," *SIAM Journal on Computing*, vol. 26, no. 3, pp. 1148–1180, 2012.

References V

[11] A. Rahman, S. Shatabda, and M. Hasan, "An Approximation Algorithm for Sorting by Reversals and Transpositions," *Journal of Discrete Algorithms*, vol. 6, no. 3, pp. 449–457, 2008.

[12] M. E. M. T. Walter, Z. Dias, and J. Meidanis, "Reversal and Transposition Distance of Linear Chromosomes," in *Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998)*, (Santa Cruz, Bolivia), pp. 96–102, IEEE Computer Society, 1998.

References VI

[13] J. Fischer and S. W. Ginzinger, "A 2-Approximation Algorithm for Sorting by Prefix Reversals," in *Proceedings of the 13th Annual European Conference on Algorithms (ESA'2005)*, (Berlin, Heidelberg), pp. 415–425, Springer-Verlag, 2005.

[14] L. Bulteau, G. Fertin, and I. Rusu, "Pancake Flipping Is Hard," in *Mathematical Foundations of Computer Science 2012*, vol. 7464 of *Lecture Notes in Computer Science*, pp. 247–258, Springer Berlin Heidelberg, 2012.

References VII

[15] D. S. Cohen and M. Blum, “On the Problem of Sorting Burnt Pancakes,” *Discrete Applied Mathematics*, vol. 61, no. 2, pp. 105–120, 1995.