

## Rearranjo de Genomas

Zanoni Dias

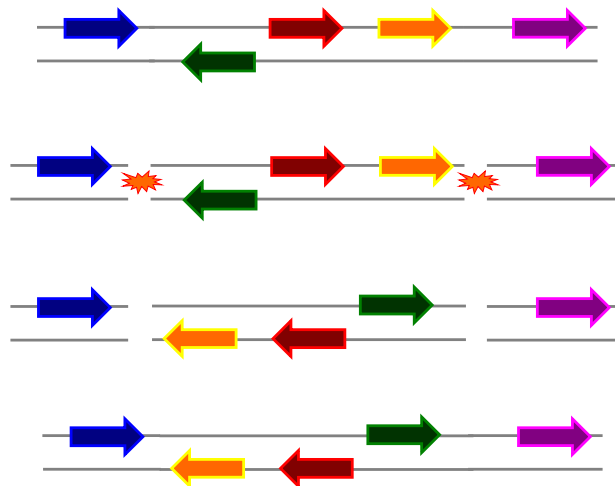
Instituto de Computação – Unicamp

6 de novembro de 2012

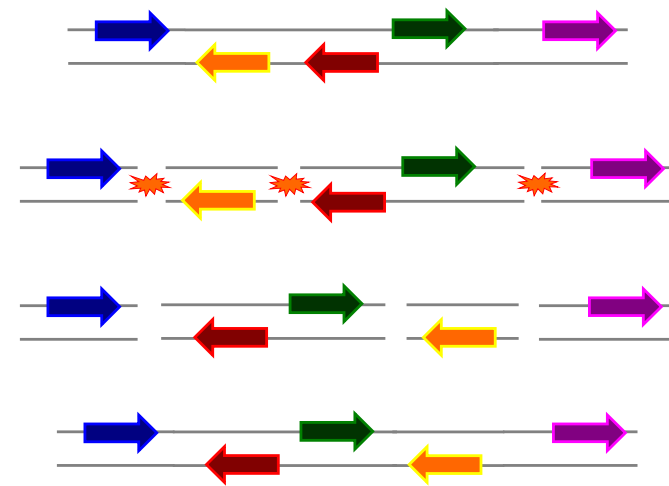
## Rearranjo de Genomas

- Comparação de genomas considerando eventos de mutação que afetam grandes porções do genoma.
- Rearranjo de genomas é uma forma mais adequada de comparar genomas completos.
- A comparação é realizada apenas com relação ao conjunto dos blocos conservados (um ou mais genes inteiros).
- Principais eventos:
  - ▶ Conservativos:
    - ★ Reversão
    - ★ Transposição
    - ★ Transposição Reversa
    - ★ Fissão
    - ★ Fusão
    - ★ Translocação
  - ▶ Não Conservativos:
    - ★ Inserção
    - ★ Remoção
    - ★ Duplicação

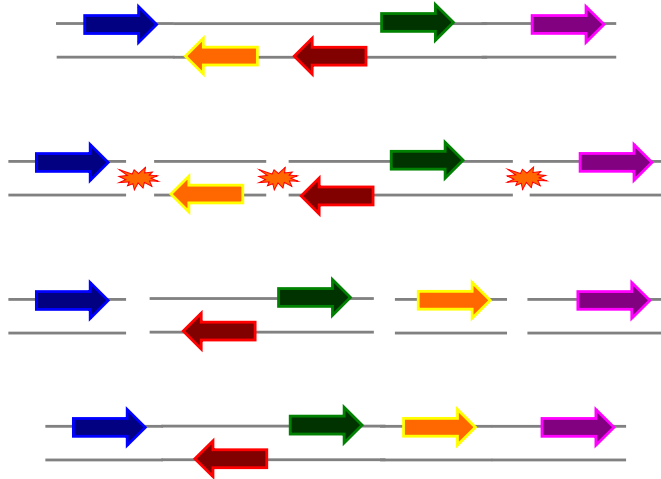
### Reversão



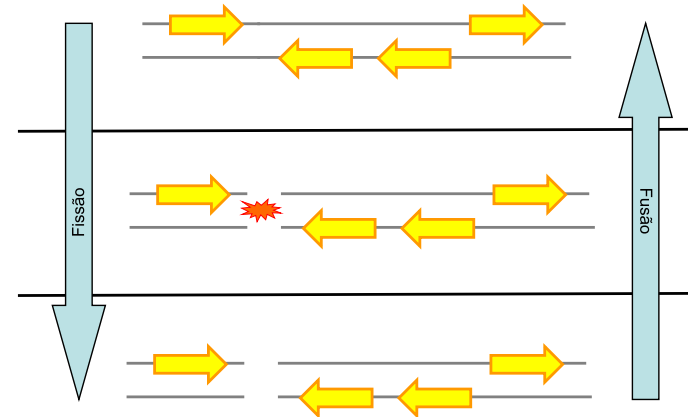
### Transposição



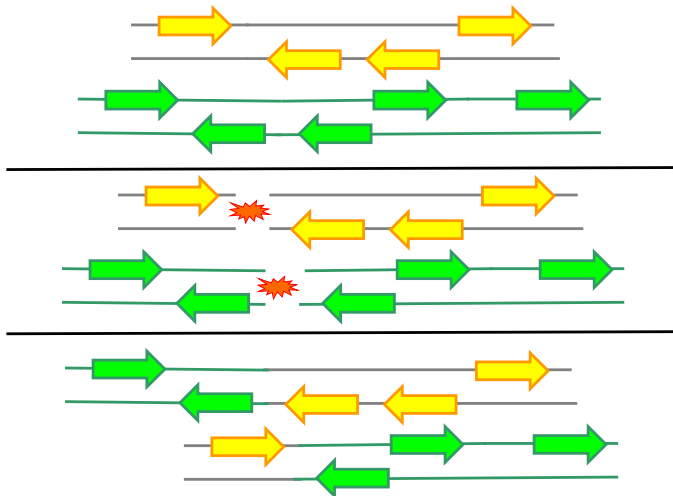
### Transposição Reversa



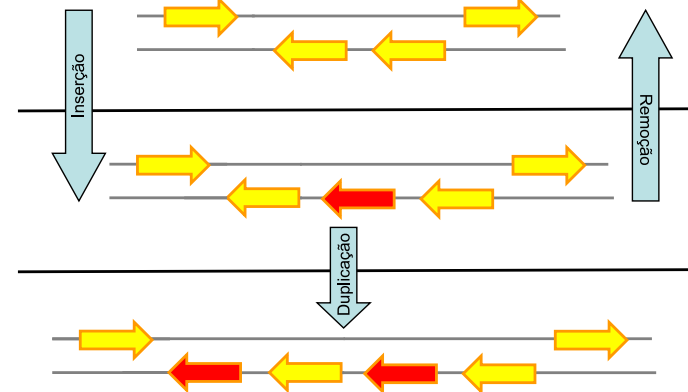
### Fissão e Fusão



### Translocação



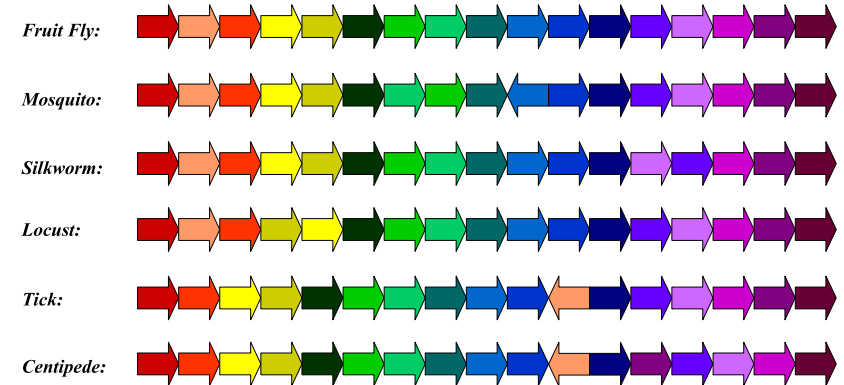
### Eventos Não Conservativos: Inserção, Remoção e Duplicação



## Genoma Mitocondrial

- Mitocôndria é uma organela envolvida no processo de respiração celular presente na maioria dos eucariotos.
- Possui um genoma circular com aproximadamente 16kbp, com 37 genes, sendo que 13 codificam proteínas, 22 codificam RNAs transportadores e 2 codificam RNAs ribossomais.
- O genoma mitocondrial é altamente conservado em animais, mas a ordem dos genes varia bastante de espécie para espécie.

## Blocos Conservados em Mitocôndrias de Artrópodes



## Blocos Conservados em Mitocôndrias de Artrópodes



## Problema da Distância de Reversão sem Orientação de Genes

- Nem sempre é possível conhecer os blocos conservados e a orientações dos genes de dois genomas.
- Podemos representar um genoma com  $n$  blocos conservados como uma permutação,  $\pi = \pi_1\pi_2 \dots \pi_n$ , dos números de 1 a  $n$ .
- A reversão  $\rho(i, j)$ , com  $1 \leq i < j \leq n$ , reverte a ordem de  $\pi[i..j]$ , ou seja,  $\pi \cdot \rho(i, j) = \pi_1\pi_2 \dots \pi_{i-1} \pi_j\pi_{j-1} \dots \pi_{i+1}\pi_i \pi_{j+1} \dots \pi_{n-1}\pi_n$ .
- *Distância de Reversão*: dados dois genomas compostos por  $n$  blocos conservados, representados pelas permutações  $\pi$  e  $\sigma$ , calcular a distância de reversão ( $d(\pi, \sigma)$ ) entre  $\pi$  e  $\sigma$ , ou seja, obter uma série de reversões  $\rho_1, \rho_2, \dots, \rho_r$ , de tamanho mínimo, tal que  $d(\pi, \sigma) = r$  e  $\pi \cdot \rho_1 \cdot \rho_2 \dots \rho_r = \sigma$ .
- *Ordenação por Reversões*: dado um genoma composto por  $n$  blocos conservados, representado pela permutação  $\pi$ , calcular a distância de reversão ( $d(\pi)$ ) entre  $\pi$  e a permutação identidade  $\iota = (1, 2, \dots, n)$ , ou seja,  $d(\pi) = d(\pi, \iota)$ .

Distância de Reversão × Ordenação por Reversões

$$\pi = 5 \quad 3 \quad 1 \quad 6 \quad 7 \quad 4 \quad 2$$

$$\sigma = 6 \quad 3 \quad 2 \quad 7 \quad 5 \quad 1 \quad 4$$

Distância de Reversão × Ordenação por Reversões

$$\pi = 5 \quad 3 \quad 1 \quad 6 \quad 7 \quad \boxed{4 \quad 2}$$

$$\sigma = 6 \quad 3 \quad 2 \quad 7 \quad 5 \quad 1 \quad 4$$

Distância de Reversão × Ordenação por Reversões

$$\pi = 5 \quad 3 \quad 1 \quad 6 \quad 7 \quad \boxed{4 \quad 2}$$

$$5 \quad 3 \quad 1 \quad 6 \quad 7 \quad 2 \quad 4$$

$$\sigma = 6 \quad 3 \quad 2 \quad 7 \quad 5 \quad 1 \quad 4$$

Distância de Reversão × Ordenação por Reversões

$$\pi = 5 \quad 3 \quad 1 \quad 6 \quad 7 \quad \boxed{4 \quad 2}$$

$$5 \quad 3 \quad \boxed{1 \quad 6 \quad 7 \quad 2} \quad 4$$

$$\sigma = 6 \quad 3 \quad 2 \quad 7 \quad 5 \quad 1 \quad 4$$

Distância de Reversão × Ordenação por Reversões

$\pi = 5 \ 3 \ 1 \ 6 \ 7 \ 4 \ 2$

5 3 1 6 7 2 4

5 3 2 7 6 1 4

$\sigma = 6 \ 3 \ 2 \ 7 \ 5 \ 1 \ 4$

Distância de Reversão × Ordenação por Reversões

$\pi = 5 \ 3 \ 1 \ 6 \ 7 \ 4 \ 2$

5 3 1 6 7 2 4

5 3 2 7 6 1 4

$\sigma = 6 \ 3 \ 2 \ 7 \ 5 \ 1 \ 4$

Distância de Reversão × Ordenação por Reversões

$\pi = 5 \ 3 \ 1 \ 6 \ 7 \ 4 \ 2$

5 3 1 6 7 2 4

5 3 2 7 6 1 4

6 7 2 3 5 1 4

$\sigma = 6 \ 3 \ 2 \ 7 \ 5 \ 1 \ 4$

Distância de Reversão × Ordenação por Reversões

$\pi = 5 \ 3 \ 1 \ 6 \ 7 \ 4 \ 2$

5 3 1 6 7 2 4

5 3 2 7 6 1 4

6 7 2 3 5 1 4

$\sigma = 6 \ 3 \ 2 \ 7 \ 5 \ 1 \ 4$

Distância de Reversão × Ordenação por Reversões

$$\pi = \begin{matrix} 5 & 3 & 1 & 6 & 7 & 4 & 2 \\ 5 & 3 & 1 & 6 & 7 & 2 & 4 \\ 5 & 3 & 2 & 7 & 6 & 1 & 4 \\ 6 & 7 & 2 & 3 & 5 & 1 & 4 \end{matrix}$$

$$\sigma = \begin{matrix} 6 & 3 & 2 & 7 & 5 & 1 & 4 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} = \iota$$

Distância de Reversão × Ordenação por Reversões

$$\pi = \begin{matrix} 5 & 3 & 1 & 6 & 7 & 4 & 2 \\ 5 & 2 & 6 & 1 & 4 & 7 & 3 \end{matrix} = \pi'$$

$$\begin{matrix} 5 & 3 & 1 & 6 & 7 & 2 & 4 \\ 5 & 3 & 2 & 7 & 6 & 1 & 4 \\ 6 & 7 & 2 & 3 & 5 & 1 & 4 \end{matrix}$$

$$\sigma = \begin{matrix} 6 & 3 & 2 & 7 & 5 & 1 & 4 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} = \iota$$

Distância de Reversão × Ordenação por Reversões

$$\pi = \begin{matrix} 5 & 3 & 1 & 6 & 7 & 4 & 2 \\ 5 & 2 & 6 & 1 & 4 & 7 & 3 \end{matrix} = \pi'$$

$$\begin{matrix} 5 & 3 & 1 & 6 & 7 & 2 & 4 \\ 5 & 2 & 6 & 1 & 4 & 3 & 7 \\ 5 & 3 & 2 & 7 & 6 & 1 & 4 \\ 6 & 7 & 2 & 3 & 5 & 1 & 4 \end{matrix}$$

$$\sigma = \begin{matrix} 6 & 3 & 2 & 7 & 5 & 1 & 4 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} = \iota$$

Distância de Reversão × Ordenação por Reversões

$$\pi = \begin{matrix} 5 & 3 & 1 & 6 & 7 & 4 & 2 \\ 5 & 2 & 6 & 1 & 4 & 7 & 3 \end{matrix} = \pi'$$

$$\begin{matrix} 5 & 3 & 1 & 6 & 7 & 2 & 4 \\ 5 & 2 & 6 & 1 & 4 & 3 & 7 \\ 5 & 3 & 2 & 7 & 6 & 1 & 4 \\ 5 & 2 & 3 & 4 & 1 & 6 & 7 \\ 6 & 7 & 2 & 3 & 5 & 1 & 4 \end{matrix}$$

$$\sigma = \begin{matrix} 6 & 3 & 2 & 7 & 5 & 1 & 4 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} = \iota$$

## Distância de Reversão × Ordenação por Reversões

$$\pi = \begin{matrix} 5 & 3 & 1 & 6 & 7 & 4 & 2 \\ 5 & 2 & 6 & 1 & 4 & 7 & 3 \end{matrix} = \pi'$$

$$\begin{matrix} 5 & 3 & 1 & 6 & 7 & 2 & 4 \\ 5 & 2 & 6 & 1 & 4 & 3 & 7 \end{matrix}$$

$$\begin{matrix} 5 & 3 & 2 & 7 & 6 & 1 & 4 \\ 5 & 2 & 3 & 4 & 1 & 6 & 7 \end{matrix}$$

$$\begin{matrix} 6 & 7 & 2 & 3 & 5 & 1 & 4 \\ 1 & 4 & 3 & 2 & 5 & 6 & 7 \end{matrix}$$

$$\sigma = \begin{matrix} 6 & 3 & 2 & 7 & 5 & 1 & 4 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} = \iota$$

### Algoritmo Ingênuo Baseado no Selection Sort para Ordenação por Reversões sem Orientação dos Genes

---

#### Algoritmo 1: Selection Sort using Reversals

---

Input:  $\pi, n$   
 $r \leftarrow 0$   
**for**  $i \leftarrow 1$  **to**  $n - 1$  **do**  
     $j \leftarrow i$   
    **while**  $\pi_j \neq i$  **do**  
         $j \leftarrow j + 1$   
    **end**  
    **if**  $j \neq i$  **then**  
         $r \leftarrow r + 1$   
         $\pi \leftarrow \pi \cdot \rho(i, j)$   
    **end**  
**end**  
**return**  $r$

---

## Ordenação por Reversões sem Orientação dos Genes

- Podemos adaptar algoritmos de ordenação para usarem apenas reversões para ordenação (sem necessariamente minimizar o número de reversões utilizadas).
- A complexidade de algoritmos de ordenação geralmente é calculada em termos do número de comparações efetuadas.
- No caso do problema da ordenação por reversões, seria interessante adaptar o algoritmo de ordenação que fizesse o menor número possível de trocas, já que as trocas de elementos devem ser transformadas em reversões.
- Entre os algoritmos de ordenação mais comumente utilizados, o Selection Sort é o único que faz no máximo  $O(n)$  trocas.

### Algoritmo Ingênuo Baseado no Selection Sort para Ordenação por Reversões sem Orientação dos Genes

- Complexidade:  $O(n^2)$ .
- Aproximação:
  - Considere a permutação  $\pi = (n, 1, 2, \dots, n - 2, n - 1)$ .
  - O algoritmo ingênuo usa  $n - 1$  reversões para ordenar  $\pi$ :
    - $\pi \leftarrow \pi \cdot \rho(1, 2) = (1, n, 2, \dots, n - 2, n - 1)$
    - $\pi \leftarrow \pi \cdot \rho(2, 3) = (1, 2, n, \dots, n - 2, n - 1)$
    - ...
    - $\pi \leftarrow \pi \cdot \rho(n - 2, n - 1) = (1, 2, \dots, n, n - 1)$
    - $\pi \leftarrow \pi \cdot \rho(n - 1, n) = (1, 2, \dots, n - 1, n) = \iota$
  - É possível ordenar  $\pi$  com apenas duas reversões:
    - $\pi \leftarrow \pi \cdot \rho(2, n) = (n, n - 1, n - 2, \dots, 2, 1)$
    - $\pi \leftarrow \pi \cdot \rho(1, n) = (1, 2, \dots, n - 1, n) = \iota$
  - Logo, o algoritmo ingênuo não garante uma aproximação melhor do que  $(n - 1)/2$ .

## O Problema da Ordenação de Panquecas

- Dada uma pilha de panquecas circulares, ordená-las, deixando a panqueca de menor diâmetro no todo da pilha. O único movimento permitido para ordenar as panquecas é o de inserir uma espátula num ponto qualquer da pilha e inverter a ordem de todas as panquecas acima da espátula.
- Qual o número mínimo de movimentos suficientes para ordenar qualquer pilha de  $n$  panquecas?
- O Problema da Ordenação de Panquecas é equivalente o problema da Ordenação por Reversões de Prefixos, ou seja, o problema da Ordenação por Reversões onde só são permitidas reversões do tipo  $\rho(1, i)$ , para  $2 \leq i \leq n$ .

## Algoritmo Guloso para o Problema de Ordenação de Panquecas

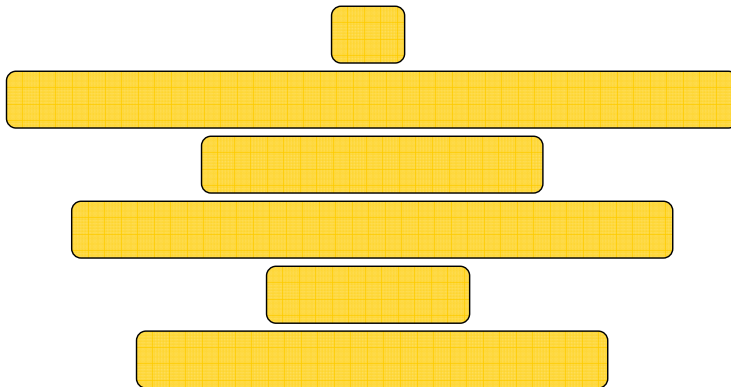
### Algoritmo 2: Greedy Pancake Flipping Problem

```

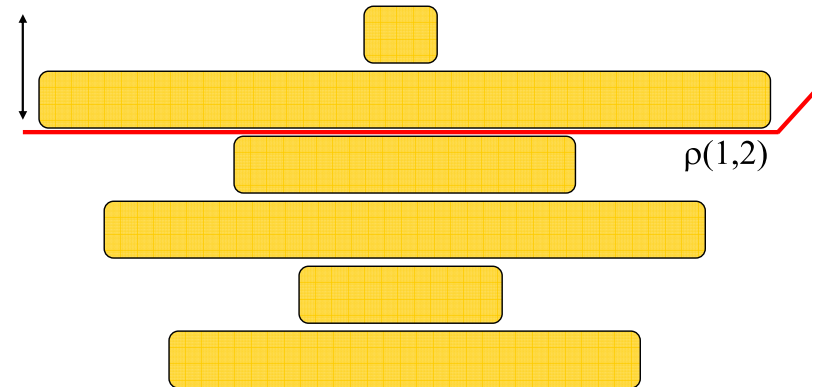
Input:  $\pi, n$ 
 $t \leftarrow 0$ 
for  $i \leftarrow n$  downto 2 do
   $j \leftarrow 1$ 
  while  $\pi_j \neq i$  do
     $j \leftarrow j + 1$ 
  end
  if  $j \neq i$  then
    if  $j \neq 1$  then
       $t \leftarrow t + 1$ 
       $\pi \leftarrow \pi \cdot \rho(1, j)$ 
    end
     $t \leftarrow t + 1$ 
     $\pi \leftarrow \pi \cdot \rho(1, i)$ 
  end
end
return  $t$ 

```

## O Problema da Ordenação de Panquecas

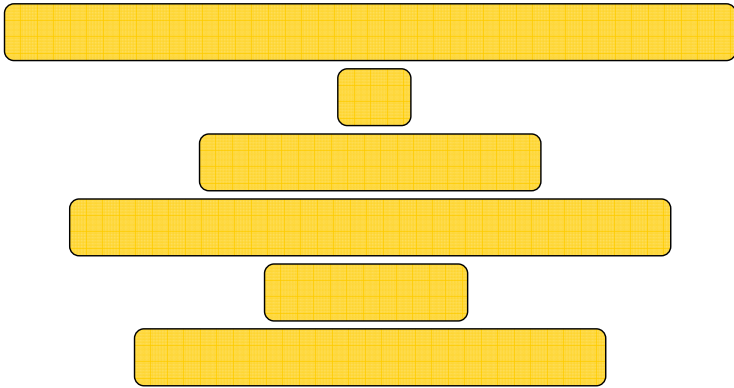


## O Problema da Ordenação de Panquecas

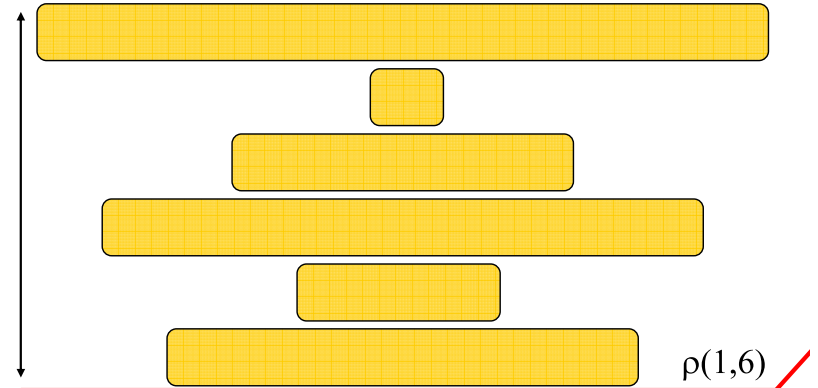




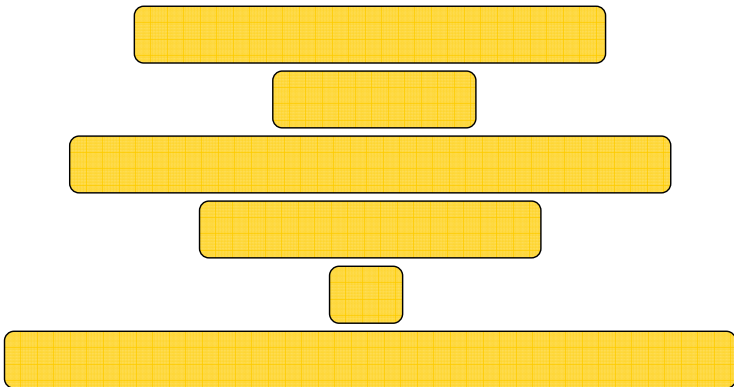
### O Problema da Ordenação de Panquecas



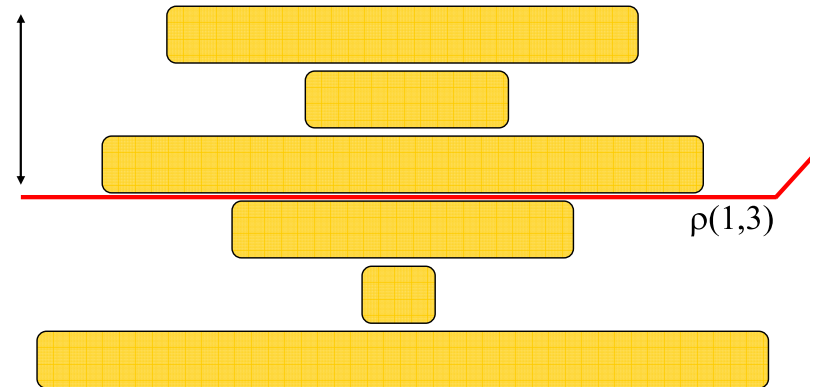
### O Problema da Ordenação de Panquecas



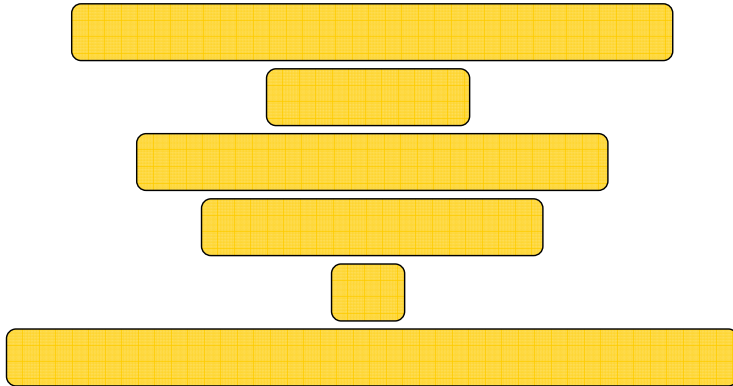
### O Problema da Ordenação de Panquecas



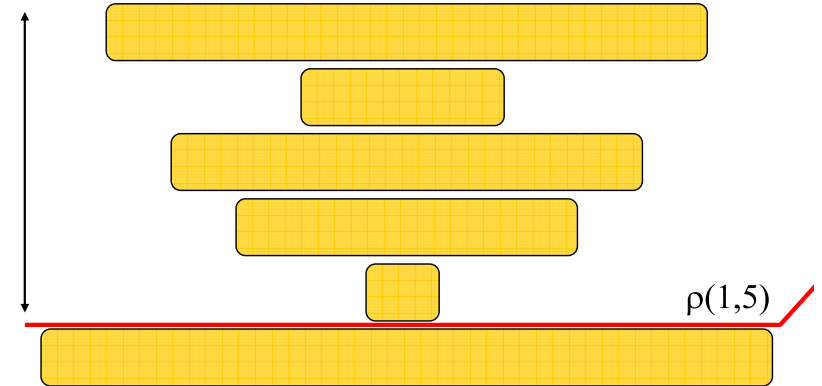
### O Problema da Ordenação de Panquecas



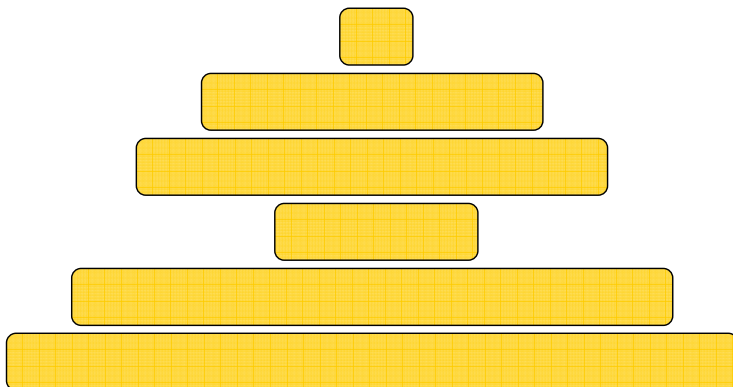
### O Problema da Ordenação de Panquecas



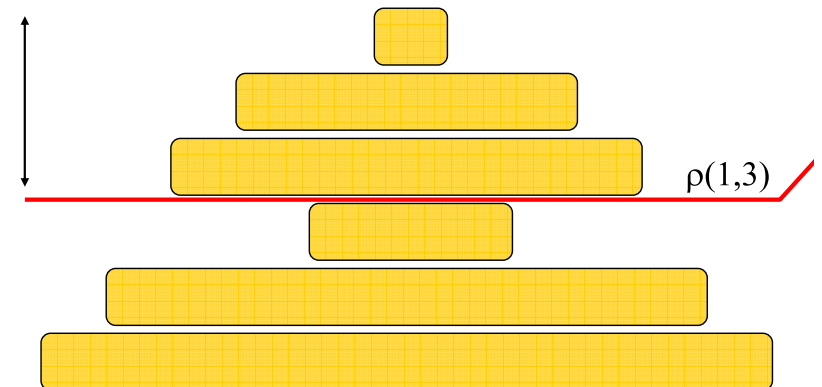
### O Problema da Ordenação de Panquecas



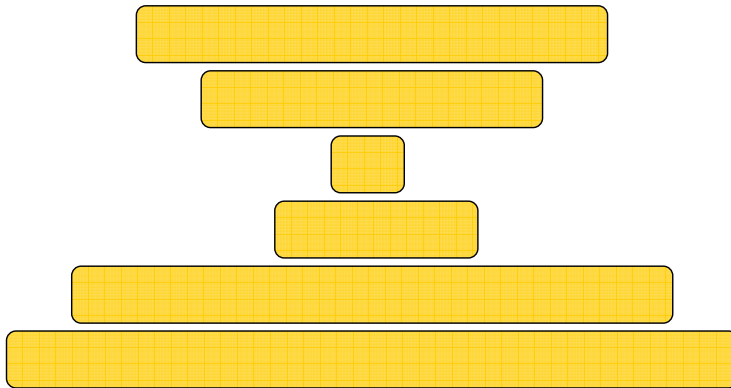
### O Problema da Ordenação de Panquecas



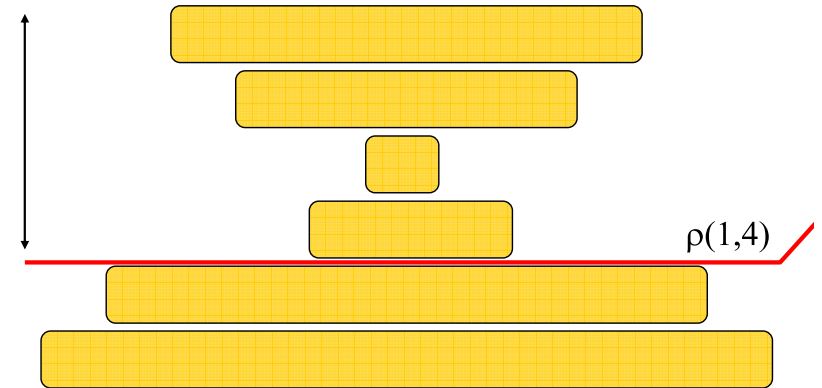
### O Problema da Ordenação de Panquecas



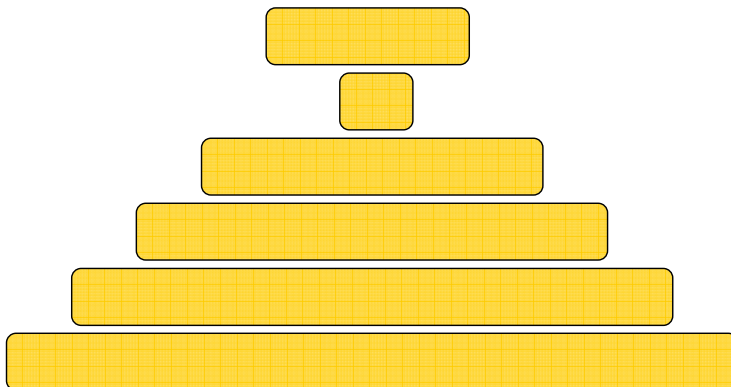
### O Problema da Ordenação de Panquecas



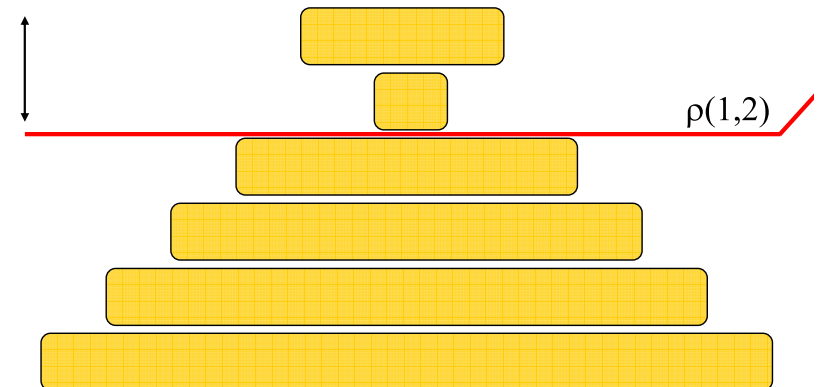
### O Problema da Ordenação de Panquecas



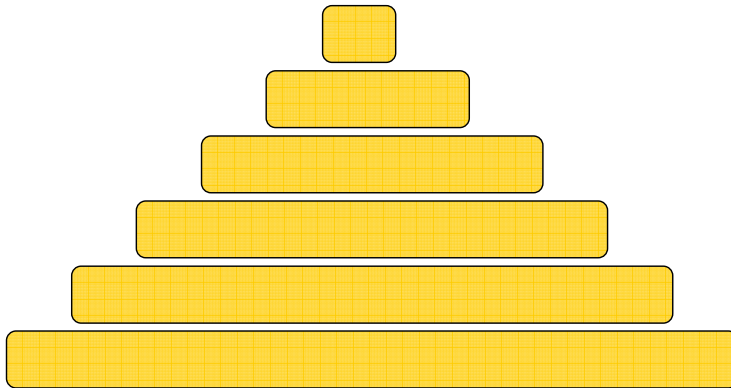
### O Problema da Ordenação de Panquecas



### O Problema da Ordenação de Panquecas



## O Problema da Ordenação de Panquecas



## O Problema da Ordenação de Panquecas

- Complexidade:  $O(n^2)$ .
- O algoritmo guloso ordena qualquer pilha de  $n$  panquecas em no máximo  $2n - 3$  movimentos.
- William Gates e Christos Papadimitriou provaram, em 1979, que  $(5n + 5)/3$  movimentos são suficientes e  $17n/16$  movimentos podem ser necessários para qualquer pilha de  $n$  panquecas
- Em 1997, Mohammad Heydari e Ivan Sudborough mostraram que podem ser necessários  $15n/14$  movimentos para ordenar uma pilha de  $n$  panquecas.
- Em 2009, Chalam Chitturi, Bill Fahle, Zhaobing Meng, Linda Morales, Charles Shields, Ivan Sudborough, e Walter Voit, pela primeira vez em 30 anos, obtiveram um limite superior melhor do que o provado por Gates e Papadimitriou: são suficientes  $(18/11)n$  movimentos para ordenar qualquer pilha de  $n$  panquecas.

## Breakpoints e Strips

- Vamos considerar a permutação estendida que pode ser obtida a partir de  $\pi$  inserindo-se dois novos elementos:  $\pi_0 = 0$  e  $\pi_{n+1} = n + 1$ .
- Um par de elementos  $\pi_i$  e  $\pi_{i+1}$ , para  $0 \leq i \leq n$ , é uma *adjacência* se  $|\pi_i - \pi_{i+1}| = 1$ . Caso contrário, o par de elementos é chamado de *breakpoint*.
- Uma *strip*  $\pi[i..j]$  é uma trecho maximal em  $\pi$  tal que todos os pares  $(\pi_k, \pi_{k+1})$  são adjacências, para  $i \leq k < j$ .
- O número de *breakpoints* numa permutação  $\pi$  é denotado por  $b(\pi)$ .
- A única permutação sem *breakpoints* é a permutação identidade ( $b(\iota) = 0$ ). Logo, ordenar por reversões é equivalente a remover todos o *breakpoints* de  $\pi$ .
- Seja  $\Delta_b(\pi, \rho) = b(\pi \cdot \rho) - b(\pi)$ . Logo,  $\Delta_b(\pi, \rho) \in \{-2, -1, 0, 1, 2\}$ .
- Podemos obter o seguinte limite inferior para o valor distância de reversão ( $d(\pi)$ ), quando a orientação dos genes é desconhecida:

$$d(\pi) \geq \frac{b(\pi)}{2}$$

## Breakpoints e Strips

	0 • 4	3 • 2 • 5	6	7 • 1 • 8	9
$\Delta_b(\pi, \rho) = +1$	0 • 4	5 • 2	3 • 6	7 • 1 • 8	9
$\Delta_b(\pi, \rho) = +2$	0 • 4	5 • 2 • 6 • 3 • 7 • 1 • 8	9		
$\Delta_b(\pi, \rho) = -1$	0 • 4	5 • 2	1 • 7 • 3 • 6 • 8	9	
$\Delta_b(\pi, \rho) = 0$	0 • 4	5 • 1	2 • 7 • 3 • 6 • 8	9	
$\Delta_b(\pi, \rho) = -2$	0 • 4	5 • 1	2	3 • 7	6 • 8 • 9

## Breakpoints e Strips

### Definição

Uma strip  $\pi[i..j]$  é chamada decrescente se e somente se a sequência  $\pi_i, \pi_{i+1}, \dots, \pi_{j-1}, \pi_j$  for decrescente. As strips unitárias são definidas como decrescentes, com exceção das strips formadas por  $\pi_0$  e  $\pi_{n+1}$  que são sempre crescentes.

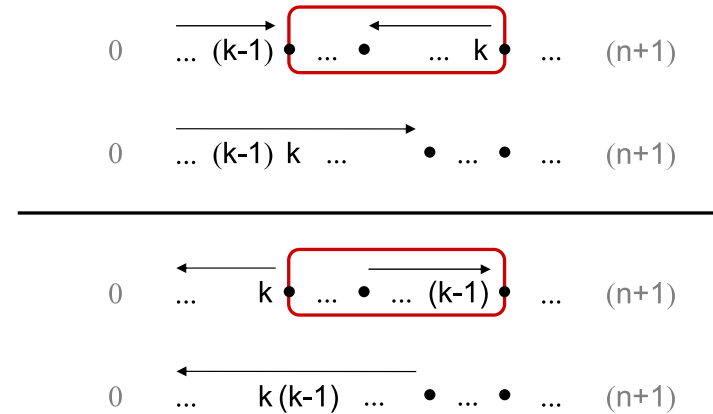
### Teorema

Se o elemento  $k$  pertence a uma strip decrescente e o elemento  $k - 1$  pertence a uma strip crescente, então existe uma reversão  $\rho$  tal que  $\Delta_b(\pi, \rho) < 0$ .

### Lema

Seja  $\pi$  uma permutação com pelo menos uma strip decrescente. Então, existe uma reversão  $\rho$  tal que  $\Delta_b(\pi, \rho) < 0$ .

## Remoção de Pelo Menos Um Breakpoint com Auxílio de uma Strip Decrescente



## Breakpoints e Strips

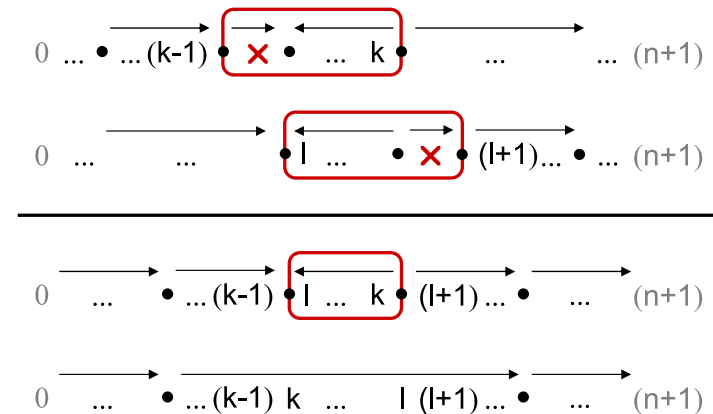
### Teorema

Seja  $\pi$  uma permutação que possui uma única strip decrescente. Se todas as reversões  $\rho$  que removem breakpoints de  $\pi$  não deixam nenhuma strip decrescente em  $\pi \cdot \rho$ , então existe uma reversão  $\rho$  tal que  $\Delta_b(\pi, \rho) = -2$ .

### Lema

Seja  $\pi$  uma permutação com pelo menos uma strip decrescente. Seja  $k$  o menor elemento entre todas as strips decrescentes de  $\pi$  e seja  $l$  o maior elemento entre todas as strips decrescentes de  $\pi$ . Seja  $\rho_k$  a reversão que posiciona  $k$  ao lado de  $k - 1$ , e seja  $\rho_l$  a reversão que posiciona  $l$  ao lado de  $l + 1$ . Se tanto  $\pi \cdot \rho_k$  quanto  $\pi \cdot \rho_l$  não possuírem nenhuma strip decrescente, então  $\rho_k = \rho_l$  e  $\Delta_b(\pi, \rho) = -2$ .

## Remoção de Dois Breakpoints ao Destruir a Última Strip Decrescente



## Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes

### Algoritmo 3: Greedy Sorting by Reversal

```

Input:  $\pi, n$ 
 $t \leftarrow 0$ 
while  $\pi \neq \iota$  do
  if  $\pi$  has a decreasing strip then
     $k \leftarrow$  the smallest element in all decreasing strips
     $\rho \leftarrow$  the reversal that cuts after  $k$  and after  $k - 1$ 
    if  $\pi \cdot \rho$  has no decreasing strip then
       $l \leftarrow$  the largest element in all decreasing strips
       $\rho \leftarrow$  the reversal that cuts before  $l$  and before  $l + 1$ 
    end
  end
  else
     $\rho \leftarrow$  the reversal that cuts the first two breakpoints
  end
   $\pi \leftarrow \pi \cdot \rho$ 
   $t \leftarrow t + 1$ 
end
return  $t$ 

```

## Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7 \ 8} \bullet \overrightarrow{5 \ 6} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$

## Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7 \ 8} \bullet \overrightarrow{5 \ 6} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$

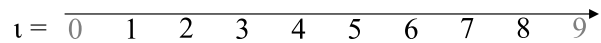
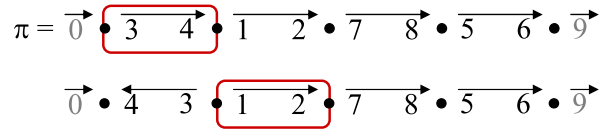
## Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7 \ 8} \bullet \overrightarrow{5 \ 6} \bullet \overrightarrow{9}$$

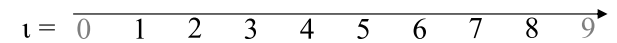
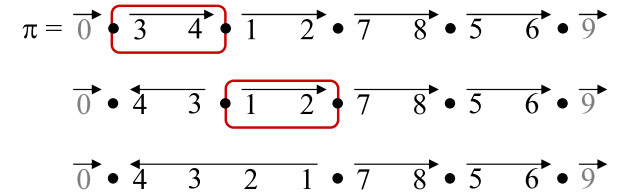
$$\overrightarrow{0} \bullet \overleftarrow{4 \ 3} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7 \ 8} \bullet \overrightarrow{5 \ 6} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$

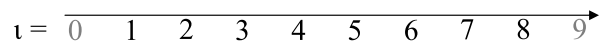
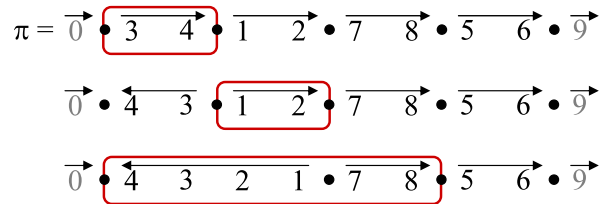
**Algoritmo Guloso para Distância de Reversão  
sem Orientação dos Genes**



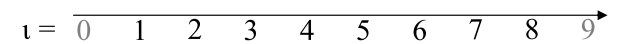
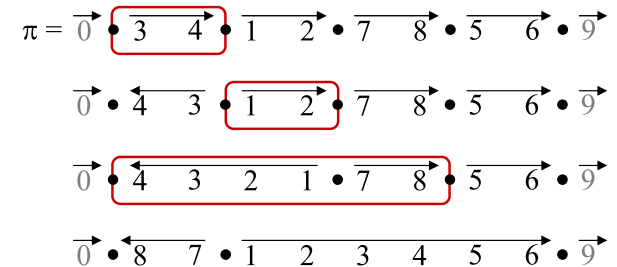
**Algoritmo Guloso para Distância de Reversão  
sem Orientação dos Genes**



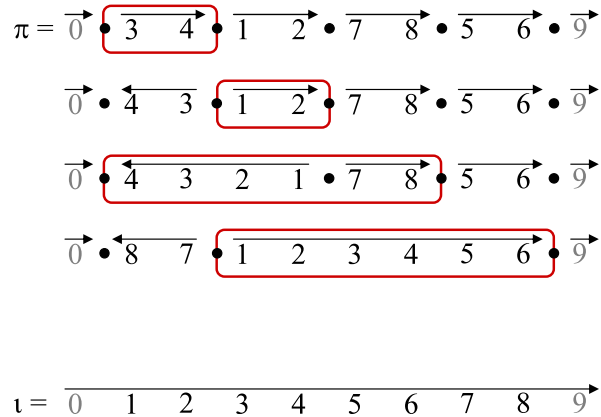
**Algoritmo Guloso para Distância de Reversão  
sem Orientação dos Genes**



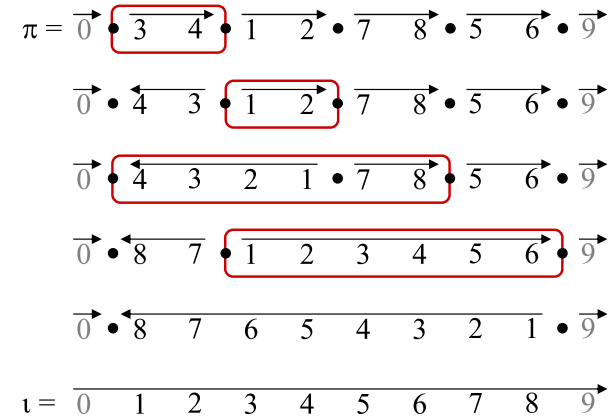
**Algoritmo Guloso para Distância de Reversão  
sem Orientação dos Genes**



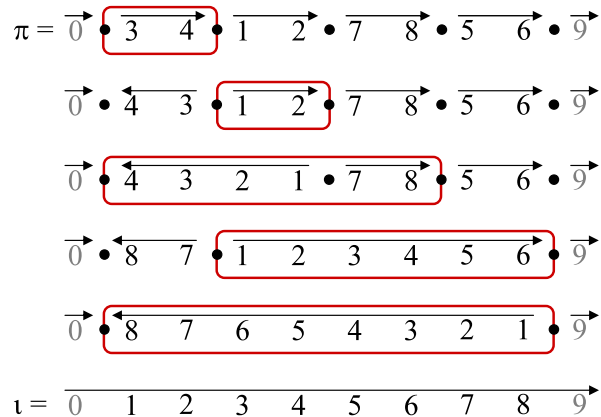
### Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes



### Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes



### Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes



### Algoritmo Guloso para Distância de Reversão sem Orientação dos Genes

- Complexidade:  $O(n^2)$ .
- O algoritmo ordena qualquer permutação usando, no máximo,  $b(\pi)$  reversões.
- Sendo assim, temos que:

$$\frac{b(\pi)}{2} \leq d(\pi) \leq b(\pi)$$

logo, o algoritmo guloso é um algoritmo de aproximação com fator:

$$\frac{b(\pi)}{\frac{b(\pi)}{2}} = 2.$$



### Ordenação Ótima por Reversões

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7 \ 8} \bullet \overrightarrow{5 \ 6} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$

### Ordenação Ótima por Reversões

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7} \bullet \boxed{\overrightarrow{8 \ 5 \ 6}} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$

### Ordenação Ótima por Reversões

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7} \bullet \boxed{\overrightarrow{8 \ 5 \ 6}} \bullet \overrightarrow{9}$$

$$\overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overleftarrow{6 \ 5 \ 8} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$

### Ordenação Ótima por Reversões

$$\pi = \overrightarrow{0} \bullet \overrightarrow{3 \ 4} \bullet \overrightarrow{1 \ 2} \bullet \overrightarrow{7} \bullet \boxed{\overrightarrow{8 \ 5 \ 6}} \bullet \overrightarrow{9}$$

$$\overrightarrow{0} \bullet \boxed{\overrightarrow{3 \ 4 \ 1}} \bullet \overrightarrow{2} \bullet \overleftarrow{6 \ 5 \ 8} \bullet \overrightarrow{9}$$

$$\iota = \overrightarrow{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9}$$



## Principais Resultados para Distância de Reversão sem Orientação dos Genes

- John Kececioğlu e David Sankoff, em 1995, apresentaram o algoritmo guloso com fator de aproximação 2 e conjecturaram que o problema de distância de reversão sem orientação é  $\mathcal{NP}$ -Completo.
- Vineet Bafna e Pavel Pevzner, em 1996, apresentaram um algoritmo com fator de aproximação 1.75.
- Alberto Caprara, em 1997, provou que o problema da distância de reversão sem orientação é  $\mathcal{NP}$ -Completo.
- David Christie, em 1998, apresentou um algoritmo com fator de aproximação 1.5.
- Piotr Berman e Marek Karpinski, em 1999, provaram que o problema da distância de reversão sem orientação é  $\mathcal{MAX-SNP}$ -Difícil.
- Piotr Berman, Sridhar Hannenhalli e Marek Karpinski, em 2002, apresentaram um algoritmo com fator de aproximação 1.375.

## Ordenação por Reversões sem Orientação dos Genes com o Número Mínimo de Reversões Imposto pelo Limite Inferior de Breakpoints

- John Kececioğlu e David Sankoff, em 1995, conjecturaram que o problema de decidir se uma permutação  $\pi$  pode ser ordenada usando exatamente  $b(\pi)/2$  reversões é um problema  $\mathcal{NP}$ -Completo.
- Nicholas Tran, em 1997, provou que é possível decidir se uma permutação  $\pi$  pode ser ordenada usando exatamente  $b(\pi)/2$  reversões, em tempo  $O(n^2 \log n)$ . O algoritmo de decisão proposto é construtivo, então, em caso afirmativo, ele exibe a sequência de reversões que ordena  $\pi$ .

## Breakpoints e Strips para Ordenação de Panquecas

- Para o problema de Ordenação de Panquecas, definimos *breakpoints* ( $b_p(\pi)$ ) e *strips* da mesma forma que para o problema da Distância de Reversão sem Orientação dos Genes, com uma única diferença:
  - ▶ O par  $(\pi_0, \pi_1)$  será sempre considerado um breakpoint, já que qualquer modificação na pilha de panquecas envolve uma “quebra” entre estas duas posições.
- Logo, a única permutação com apenas um *breakpoint* é a permutação identidade ( $b_p(\iota) = 1$ ).
- Seja  $\Delta_{b_p}(\pi, \rho) = b_p(\pi \cdot \rho) - b_p(\pi)$ . Então,  $\Delta_{b_p}(\pi, \rho) \in \{-1, 0, 1\}$ .
- É possível obter um limite inferior para o número de movimentos necessários para ordenar uma pilha de panquecas ( $d_p(\pi)$ ), com base no número de *breakpoints* de uma permutação  $\pi$ :

$$d_p(\pi) \geq b_p(\pi) - 1$$

## Algoritmos de Aproximação para Ordenação de Panquecas

### Exercício

Mostre que o algoritmo guloso para o problema de Ordenação de Panquecas é um algoritmo de aproximação com fator 4.

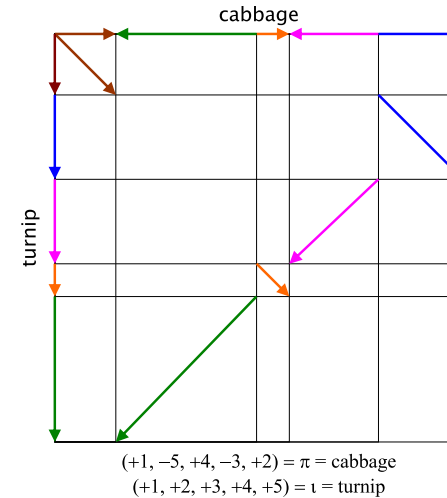
### Exercício

Mostre um algoritmo de aproximação com fator 3 para o problema de Ordenação de Panquecas.

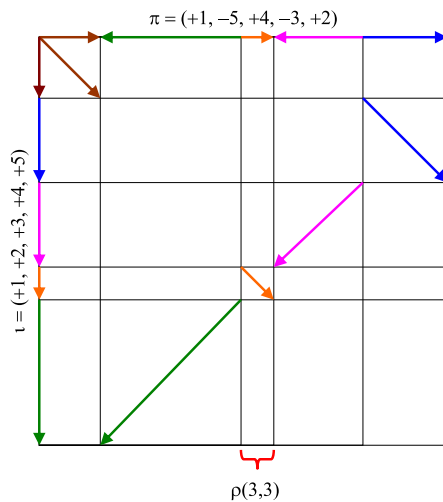
## Problema da Distância de Reversão com Orientação de Genes

- Podemos representar um genoma com  $n$  blocos conservados com orientação dos genes conhecida como uma permutação sinalizada,  $\pi = \pi_1 \pi_2 \dots \pi_n$ , com  $\pi_i \in \{-1, -2, \dots, -n, +1, +2, \dots, +n\}$ , de tal forma que  $|\pi_i| - |\pi_j| \neq 0$ , para  $1 \leq i < j \leq n$ .
- A reversão  $\rho(i, j)$ , com  $1 \leq i \leq j \leq n$ , reverte a ordem de  $\pi[i..j]$  e os sinais de todos os elementos pertencentes a este intervalo.
- Distância de Reversão:** dados dois genomas compostos por  $n$  blocos conservados, representados pelas permutações sinalizadas  $\pi$  e  $\sigma$ , calcular a distância de reversão ( $d(\pi, \sigma)$ ) entre  $\pi$  e  $\sigma$ , ou seja, obter uma série de reversões  $\rho_1, \rho_2, \dots, \rho_r$ , de tamanho mínimo, tal que  $d(\pi, \sigma) = r$  e  $\pi \cdot \rho_1 \cdot \rho_2 \dots \rho_r = \sigma$ .
- Ordenação por Reversões:** dado um genoma composto por  $n$  blocos conservados, representado pela permutação sinalizada  $\pi$ , calcular a distância de reversão ( $d(\pi)$ ) entre  $\pi$  e a permutação identidade  $\iota = (+1, +2, \dots, +n)$ , ou seja,  $d(\pi) = d(\pi, \iota)$ .

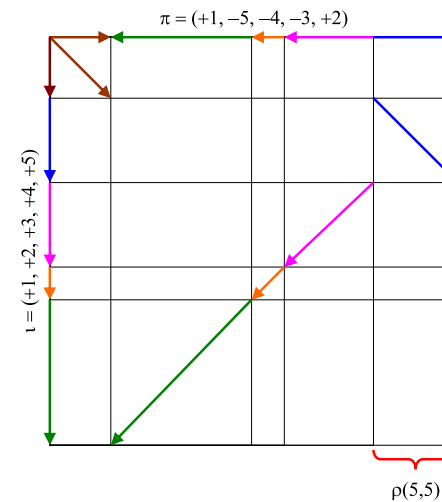
## Transforming Cabbage into Turnip



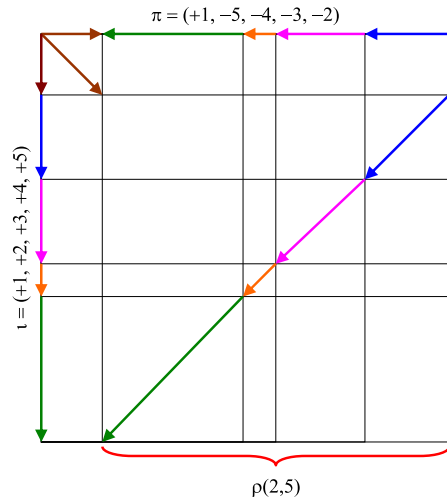
## Transforming Cabbage into Turnip



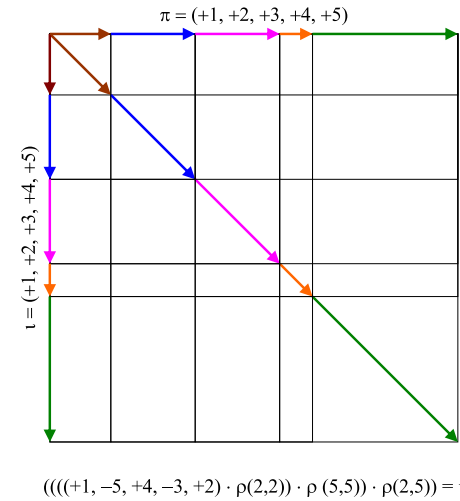
## Transforming Cabbage into Turnip



## Transforming Cabbage into Turnip



## Transforming Cabbage into Turnip



$$(((+1, -5, +4, -3, +2) \cdot \rho(2,2)) \cdot \rho(5,5)) \cdot \rho(2,5) = \tau$$

## Breakpoints e Permutação Reduzida

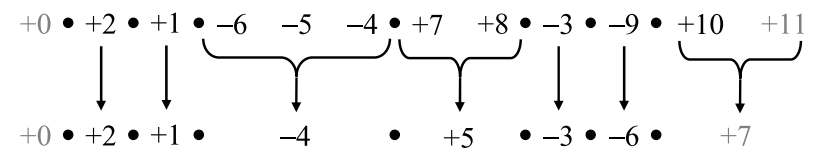
### Definição

Um par de elementos  $\pi_i$  e  $\pi_{i+1}$ , para  $0 \leq i \leq n$ , é uma adjacência se  $\pi_{i+1} - \pi_i = 1$ . Caso contrário, o par de elementos é chamado de breakpoint.

### Definição

Uma permutação  $\pi$  é chamada reduzida se ela não contém adjacências.

## Permutação Reduzida



## Reversão Orientada

### Definição

Um par orientado  $(\pi_i, \pi_j)$  é um par de elementos de  $\pi$ , tal que  $i < j$ ,  $|\pi_i| - |\pi_j| = 1$  e  $\pi_i \times \pi_j < 0$ .

### Definição

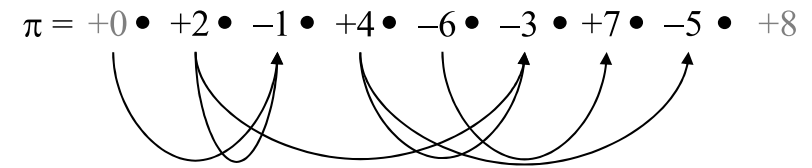
A pontuação de uma reversão  $\rho$  em relação a  $\pi$ , representada por  $\text{score}(\pi, \rho)$ , é o número de pares orientados em  $\pi \cdot \rho$ .

### Definição

Seja  $(\pi_i, \pi_j)$  um par orientado. Logo, as seguintes reversões são chamadas orientadas:

- $\rho(i, j - 1)$ , se  $\pi_i + \pi_j = +1$ .
- $\rho(i + 1, j)$ , se  $\pi_i + \pi_j = -1$ .

## Pares Orientados × Reversões Orientadas



$$(+0, -1) \Rightarrow \rho(1,2)$$

$$(+4, -3) \Rightarrow \rho(3,4)$$

$$(+2, -1) \Rightarrow \rho(1,1)$$

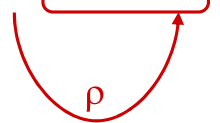
$$(+4, -5) \Rightarrow \rho(4,7)$$

$$(+2, -3) \Rightarrow \rho(2,5)$$

$$(-6, +7) \Rightarrow \rho(4,5)$$

## Pares Orientados × Reversões Orientadas

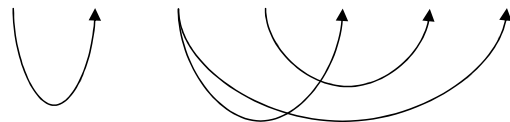
$\pi = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$



$$(+0, -1) \Rightarrow \rho(1,2)$$

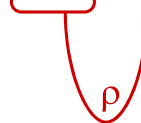
$$\text{score}(\pi, \rho) = 4$$

$\pi \cdot \rho = +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$



## Pares Orientados × Reversões Orientadas

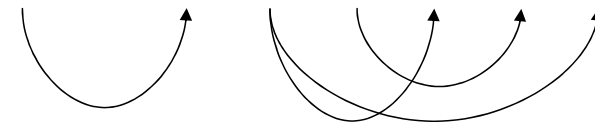
$\pi = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$



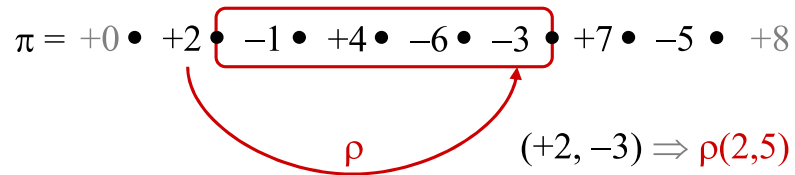
$$(+2, -1) \Rightarrow \rho(1,1)$$

$$\text{score}(\pi, \rho) = 4$$

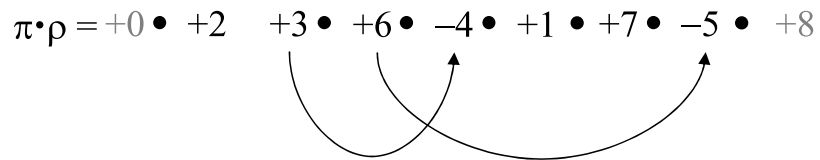
$\pi \cdot \rho = +0 \bullet -2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$



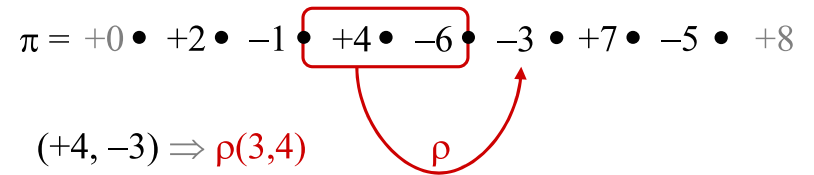
Pares Orientados × Reversões Orientadas



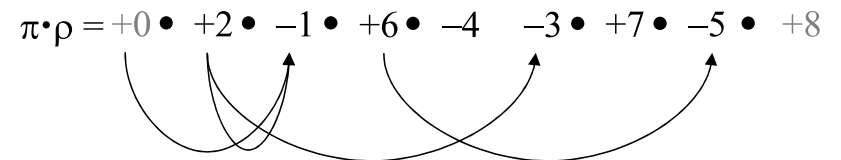
$\text{score}(\pi, \rho) = 2$



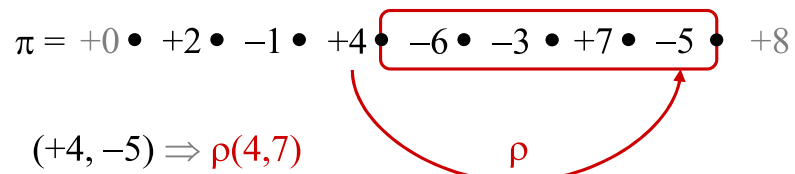
Pares Orientados × Reversões Orientadas



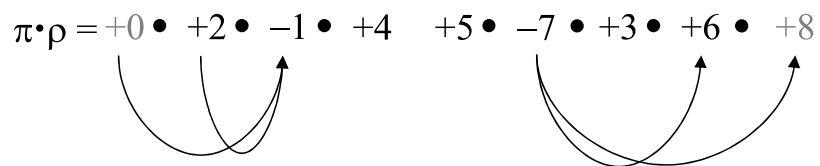
$\text{score}(\pi, \rho) = 4$



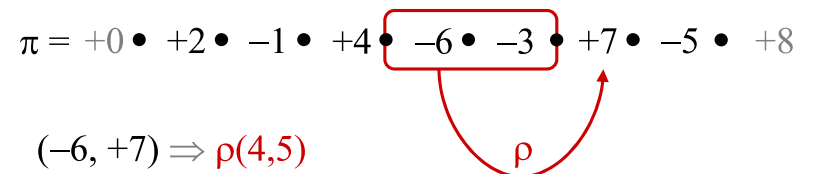
Pares Orientados × Reversões Orientadas



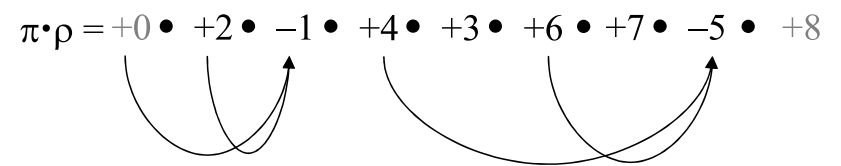
$\text{score}(\pi, \rho) = 4$



Pares Orientados × Reversões Orientadas



$\text{score}(\pi, \rho) = 4$



## Ordenação Usando Apenas Pares Ordenados

$$\pi^{(0)} = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

### Pares Orientados $\times$ Reversões Orientadas

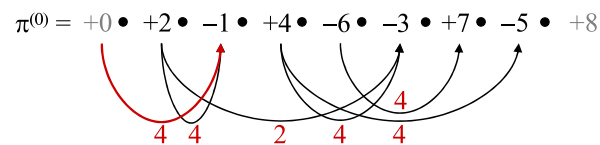
#### Lema

Uma reversão  $\rho$  é orientada em relação a  $\pi$  se e somente se  $\Delta_b(\pi, \rho) < 0$ .

#### Teorema

Seja  $\pi^{(i)}$  uma permutação que contém pelo menos um par orientado. Seja  $\rho_i$  uma reversão orientada de score máximo em relação a  $\pi^{(i)}$ . Defina  $\pi^{(i+1)}$  como  $\pi^{(i+1)} = (\pi^{(i)} \cdot \rho_i)$ . Seja  $\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(k)}$  uma série maximal de permutações gerada a partir de  $\pi^{(0)}$ . Logo  $\pi^{(k)}$  é formada apenas por elementos positivos e  $d(\pi^{(0)}, \pi^{(k)}) = k$ .

## Ordenação Usando Apenas Pares Ordenados



## Ordenação Usando Apenas Pares Ordenados

$$\pi^{(0)} = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$



### Ordenação Usando Apenas Pares Ordenados

$$\pi^{(0)} = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

$$\pi^{(1)} = +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

### Ordenação Usando Apenas Pares Ordenados

$$\pi^{(0)} = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

$$\pi^{(1)} = +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

### Ordenação Usando Apenas Pares Ordenados

$$\pi^{(0)} = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

$$\pi^{(1)} = +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

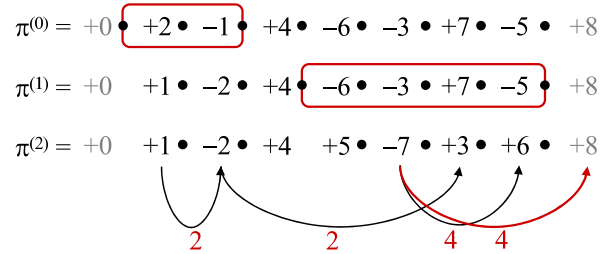
### Ordenação Usando Apenas Pares Ordenados

$$\pi^{(0)} = +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

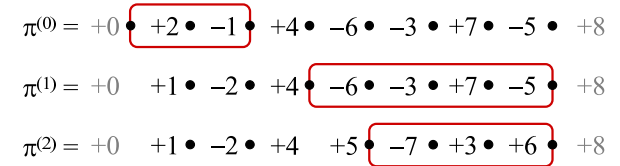
$$\pi^{(1)} = +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8$$

$$\pi^{(2)} = +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -7 \bullet +3 \bullet +6 \bullet +8$$

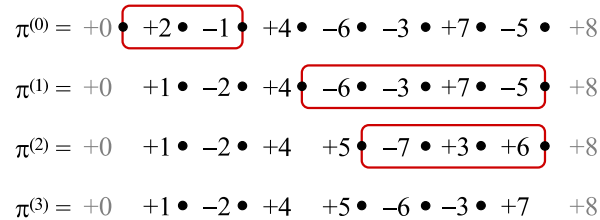
### Ordenação Usando Apenas Pares Ordenados



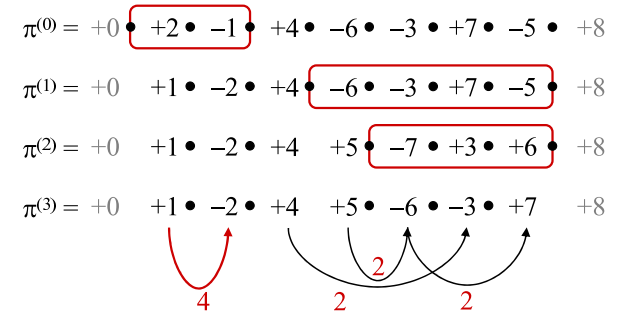
### Ordenação Usando Apenas Pares Ordenados



### Ordenação Usando Apenas Pares Ordenados



### Ordenação Usando Apenas Pares Ordenados



### Ordenação Usando Apenas Pares Ordenados

$$\begin{aligned} \pi^{(0)} &= +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(1)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(2)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -7 \bullet +3 \bullet +6 \bullet +8 \\ \pi^{(3)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \end{aligned}$$

### Ordenação Usando Apenas Pares Ordenados

$$\begin{aligned} \pi^{(0)} &= +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(1)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(2)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -7 \bullet +3 \bullet +6 \bullet +8 \\ \pi^{(3)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \\ \pi^{(4)} &= +0 \bullet +1 \bullet +2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \end{aligned}$$

### Ordenação Usando Apenas Pares Ordenados

$$\begin{aligned} \pi^{(0)} &= +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(1)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(2)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -7 \bullet +3 \bullet +6 \bullet +8 \\ \pi^{(3)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \\ \pi^{(4)} &= +0 \bullet +1 \bullet +2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \end{aligned}$$

### Ordenação Usando Apenas Pares Ordenados

$$\begin{aligned} \pi^{(0)} &= +0 \bullet +2 \bullet -1 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(1)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet -6 \bullet -3 \bullet +7 \bullet -5 \bullet +8 \\ \pi^{(2)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -7 \bullet +3 \bullet +6 \bullet +8 \\ \pi^{(3)} &= +0 \bullet +1 \bullet -2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \\ \pi^{(4)} &= +0 \bullet +1 \bullet +2 \bullet +4 \bullet +5 \bullet -6 \bullet -3 \bullet +7 \bullet +8 \end{aligned}$$





## Hurdles

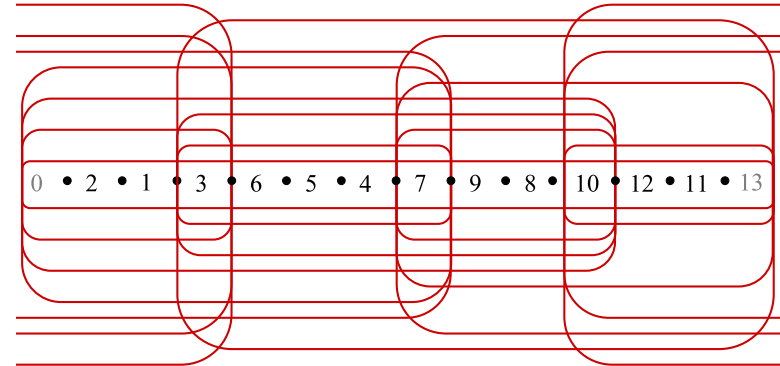
### Definição

Seja  $\pi$  uma permutação reduzida formada apenas por elementos positivos (logo, sem pares orientados). Suponha que  $\pi$  foi estendida, com  $\pi_0 = 0$  e  $\pi_{n+1} = n + 1$ , e circularizada, considerando que o elemento 0 é consecutivo ao elemento  $n + 1$ . Um framed interval em  $\pi$  é um intervalo da forma  $i \pi_{j+1} \pi_{j+2} \dots \pi_{j+k-1} i + k$ , tal que todos inteiros entre  $i$  e  $i + k$  pertencem ao intervalo  $[i..i + k]$  (considerado de forma circular).

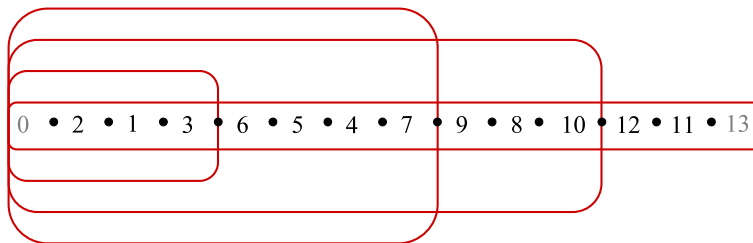
### Definição

Seja  $\pi$  uma permutação reduzida formada apenas por elementos positivos. Um hurdle em  $\pi$  é um framed interval que não contém outros framed intervals.

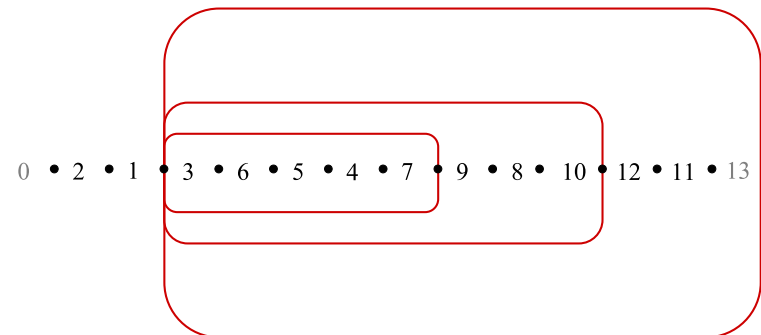
## Framed Intervals



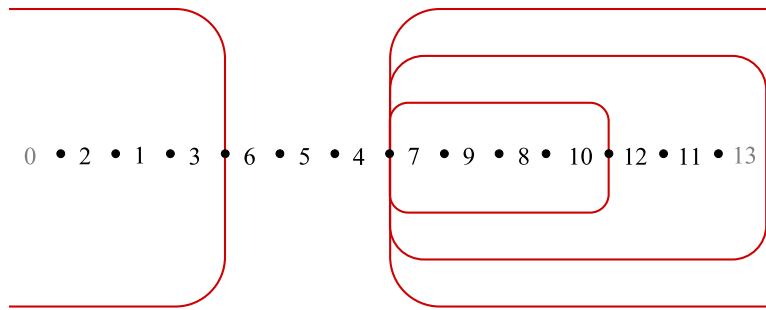
## Framed Intervals



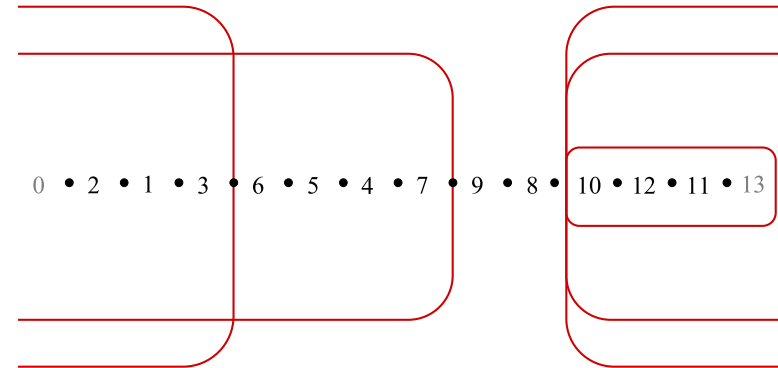
## Framed Intervals



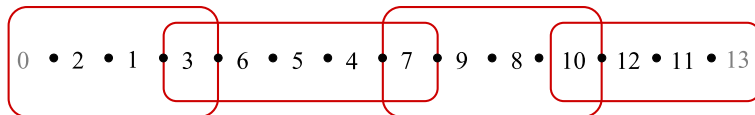
## Framed Intervals



## Framed Intervals



## Hurdles



## Hurdles

### Definição

A posição do elemento  $j$  na permutação  $\pi$  é indicada por  $\pi_j^{-1}$ .

### Definição

Uma reversão  $\rho$  corta um hurdle  $i \pi_{j+1} \pi_{j+2} \dots i+1 \dots \pi_{j+k-1} i+k$  se  $\rho = \rho(\pi_i^{-1} + 1, \pi_{i+1}^{-1} - 1)$ , ou seja, se reverte os elementos entre  $i$  e  $i+1$ .

### Definição

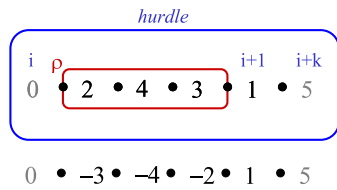
Uma reversão  $\rho$  une dois hurdles  $i \dots i+k \dots i' \dots i'+k'$  da permutação  $\pi$  se  $\rho = \rho(\pi_{i+k}^{-1}, \pi_{i'}^{-1})$ , ou seja, se reverte os elementos entre  $i+k$  e  $i'$  (inclusive ambos).

### Definição

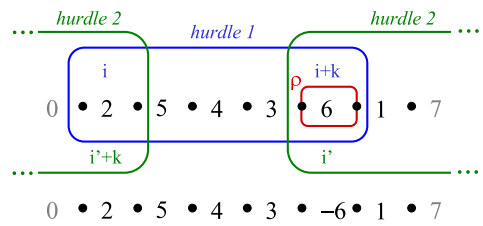
Um hurdle é chamado de simples se quando cortado o número de hurdles diminui. Caso contrário, o hurdle é chamado de super.

### Cutting Hurdles × Merging Hurdles

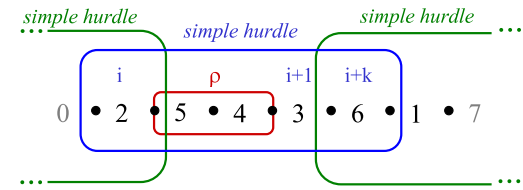
Cutting Hurdles



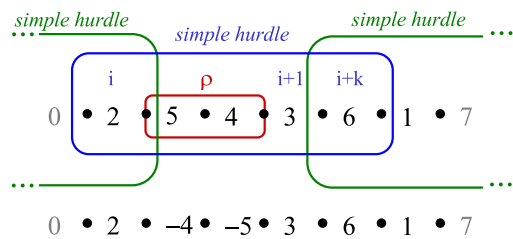
Merging Hurdles



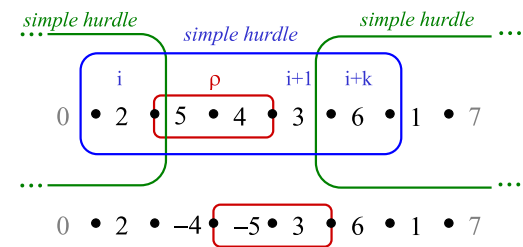
### Simple Hurdle



### Simple Hurdle

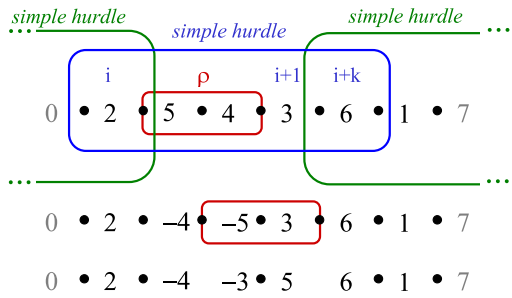


### Simple Hurdle

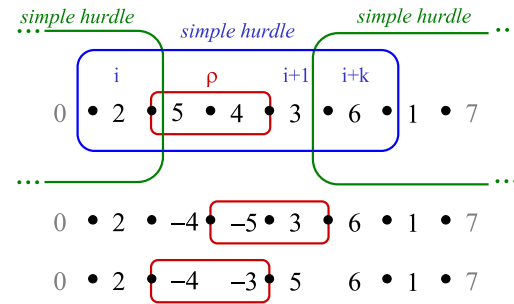




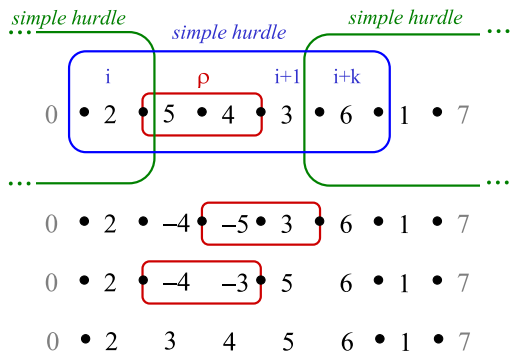
### Simple Hurdle



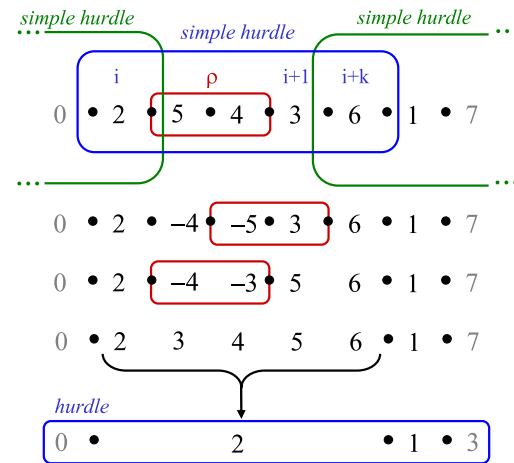
### Simple Hurdle



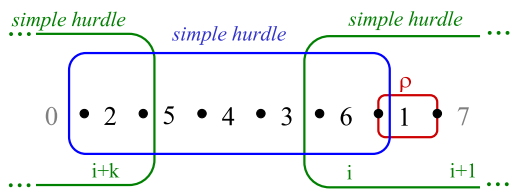
### Simple Hurdle



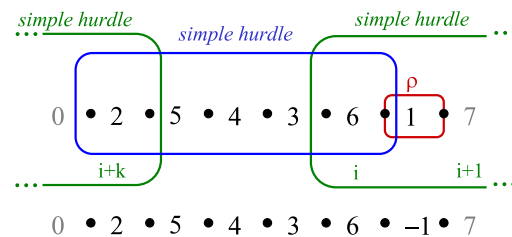
### Simple Hurdle



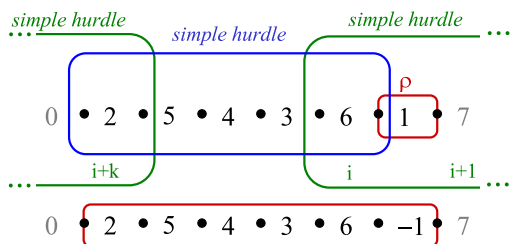
### Simple Hurdle



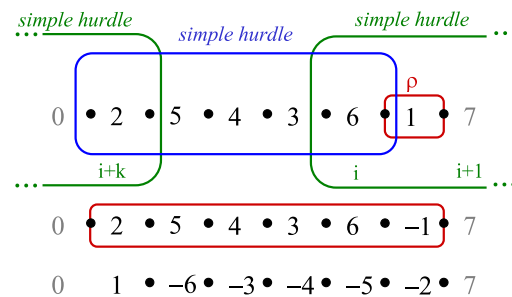
### Simple Hurdle



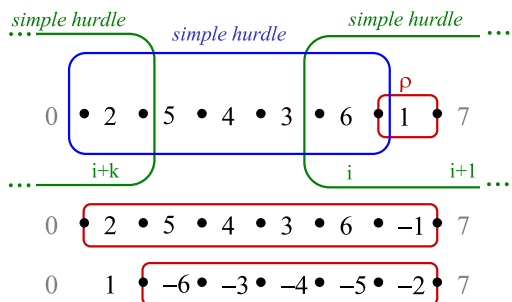
### Simple Hurdle



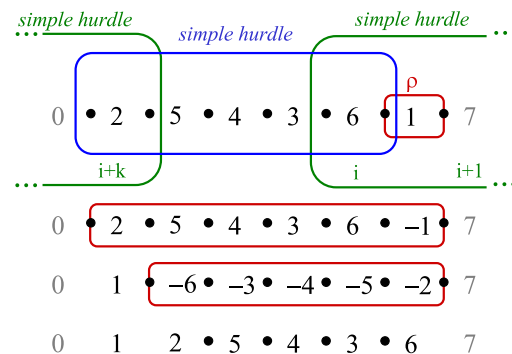
### Simple Hurdle



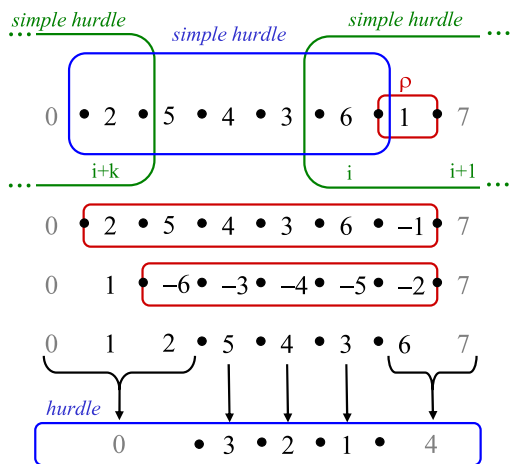
### Simple Hurdle



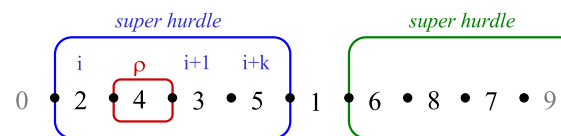
### Simple Hurdle



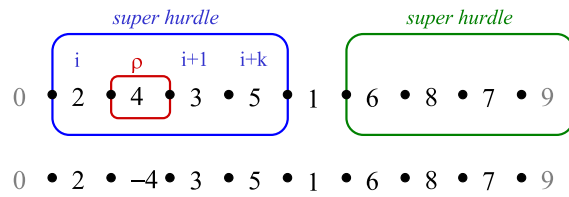
### Simple Hurdle



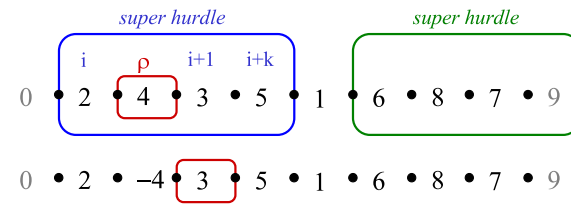
### Super Hurdle



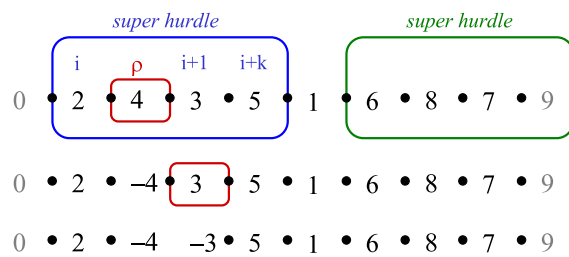
### Super Hurdle



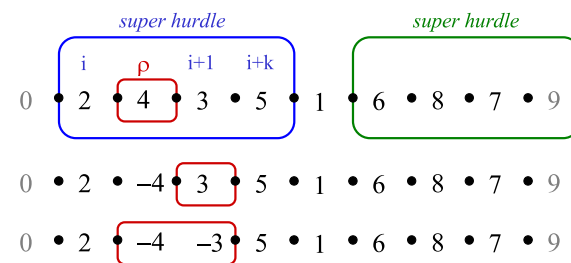
### Super Hurdle



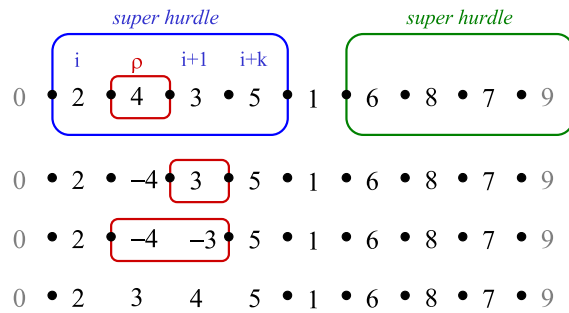
### Super Hurdle



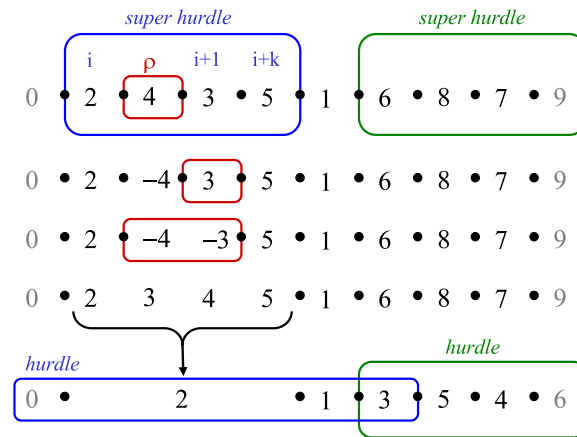
### Super Hurdle



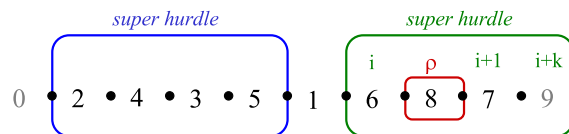
### Super Hurdle



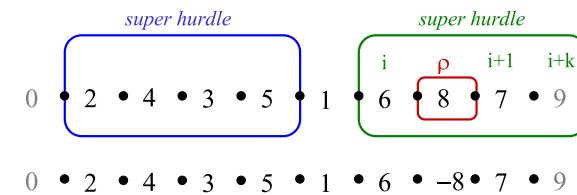
### Super Hurdle



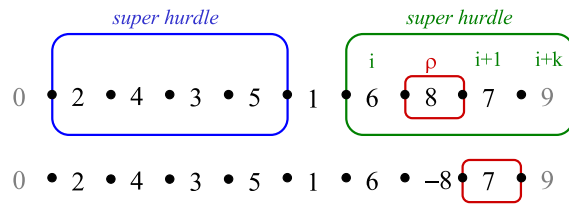
### Super Hurdle



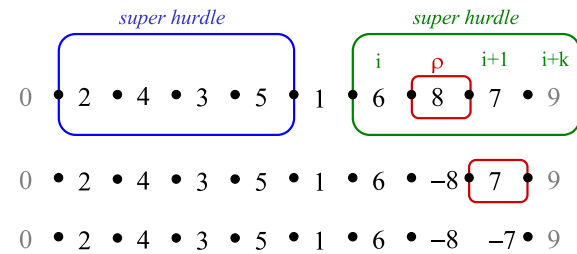
### Super Hurdle



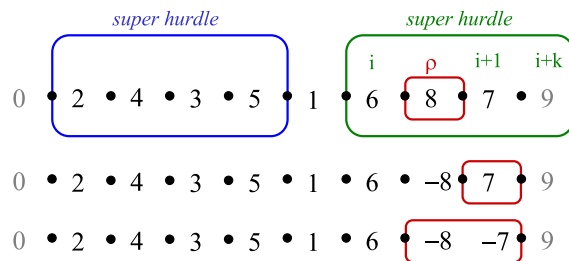
### Super Hurdle



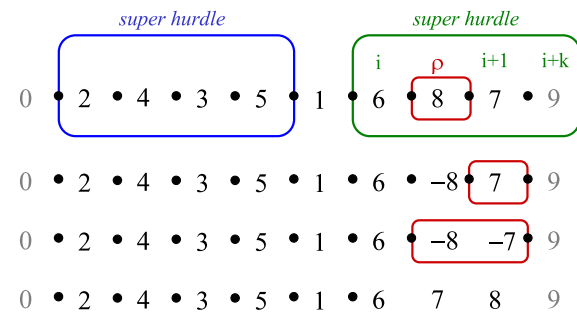
### Super Hurdle



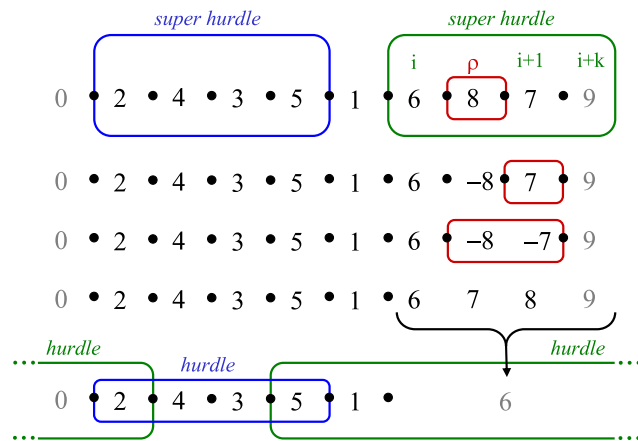
### Super Hurdle



### Super Hurdle



## Super Hurdle



## Algoritmo Ótimo para Distância de Reversão com Orientação dos Genes

### Algoritmo 4: Optimal Sorting by Reversal

```

Input:  $\pi, n$ 
 $t \leftarrow 0$ 
while  $\pi \neq \iota$  do
  if  $\pi$  has a oriented pair then  $\rho \leftarrow$  the reversal that has maximal score else
    if  $\pi$  has 2k hurdles then
      if  $\pi$  has 2 hurdles then  $\rho \leftarrow$  the reversal that merges the two hurdles
      else  $\rho \leftarrow$  any reversal that merges two non-consecutives hurdles
    end
    if  $\pi$  has only one hurdle then  $\rho \leftarrow$  the reversal that cuts the hurdle else
      if  $\pi$  has a simple hurdle then  $\rho \leftarrow$  any reversal that cuts a simple hurdle else
        if  $\pi$  has 3 hurdles then  $\rho \leftarrow$  any reversal that merges two hurdles
        else  $\rho \leftarrow$  any reversal that merges two non-consecutives hurdles
      end
    end
  end
   $\pi \leftarrow \pi \cdot \rho$ 
   $t \leftarrow t + 1$ 
end
return t

```

## Algoritmo Ótimo para Distância de Reversão com Orientação dos Genes

- Complexidade:
  - Determinar todos os pares orientados de uma permutação:  $O(n)$ .
  - Determinar a reversão com maior *score*:  $O(n) \times O(n) = O(n^2)$ .
  - Determinar todos os *hurdles*:  $O(n^2)$ .
  - Total:  $O(n) \times O(n^2) = O(n^3)$ .
- Algoritmo proposto por Anne Bergeron, em 2004.
- O problema da distância de reversão com orientação de genes foi originalmente resolvido em tempo polinomial ( $O(n^4)$ ) por Sridhar Hannenhalli e Pavel Pevzner, em 1999.
- David Bader, Bernard Moret e Mi Yan, em 2001, mostraram que é possível calcular a distância de reversão com orientação de genes conhecida (sem listar as reversões utilizadas), em  $O(n)$ .
- Krister Swenson, Yu Lin, Vaibhav Rajan e Bernard Moret, em 2008, provaram que a chance de uma permutação aleatória (com sinal) possuir pelo menos um *hurdle* é de  $\Theta(n^{-2})$ .

## Distância de Reversão, Translocação, Fusão e Fissão

- Sridhar Hannenhalli e Pavel Pevzner, em 1999, apresentaram o primeiro algoritmo polinomial para este problema.
- Pavel Pevzner e Glenn Tesler, em 2003, mostraram um cenário completo de evolução entre humanos e camundongos:
  - Eles consideraram 281 blocos conservados com pelo menos 1 Mbp.
  - Os blocos conservados no genoma humano tem tamanho médio de 9.6 Mbp, enquanto no camundongo, possuem tamanho médio de 8.5 Mbp.
  - Os blocos conservados no genoma humano cobrem 2707 Mbp (94.0% do genoma), enquanto no camundongo cobrem 2397 Mbp (95.3%).
  - Os *breakpoints* no genoma humano tem tamanho médio de 668 kbp enquanto no camundongo, possuem tamanho médio de 458 kbp.
  - Existe um cenário ótimo de evolução envolvendo 245 eventos (149 reversões, 93 translocações e 3 fissões).
  - Existem outros cenários possíveis com 245 eventos (o cenário anterior é o que apresenta o maior número de reversões).
  - Foram detectados também 3170 microarranjos (reversões), dentro dos blocos conservados.

Problema da Distância de Transposição

- A transposição  $\rho(i, j, k)$ , com  $1 \leq i < j < k \leq n + 1$ , troca os blocos  $\pi[i..j - 1]$  e  $\pi[j..k - 1]$  de lugar, ou seja,  $\pi \cdot \rho(i, j, k) = \pi_1 \pi_2 \dots \pi_{i-1} \pi_j \pi_{j+1} \dots \pi_{k-i} \pi_i \pi_{i+1} \dots \pi_{j-1} \pi_k \dots \pi_n$ .
- *Distância de Transposições*: dados dois genomas compostos por  $n$  blocos conservados, representados pelas permutações  $\pi$  e  $\sigma$ , calcular a distância de transposição ( $d_t(\pi, \sigma)$ ) entre  $\pi$  e  $\sigma$ , ou seja, obter uma série de transposições  $\rho_1, \rho_2, \dots, \rho_t$ , de tamanho mínimo, tal que  $d_t(\pi, \sigma) = t$  e  $\pi \cdot \rho_1 \cdot \rho_2 \dots \rho_t = \sigma$ .
- *Ordenação por Transposições*: dado um genoma composto por  $n$  blocos conservados, representado pela permutação  $\pi$ , calcular a distância de transposição ( $d_t(\pi)$ ) entre  $\pi$  e a permutação identidade  $\iota = (1, 2, \dots, n)$ , ou seja,  $d_t(\pi) = d_t(\pi, \iota)$ .

- Vamos considerar a permutação estendida que pode ser obtida a partir de  $\pi$  inserindo-se dois novos elementos:  $\pi_0 = 0$  e  $\pi_{n+1} = n + 1$ .
- Um par de elementos  $\pi_i$  e  $\pi_{i+1}$ , para  $0 \leq i \leq n$ , é uma *adjacência* se  $\pi_{i+1} - \pi_i = 1$ . Caso contrário, o par de elementos é chamado de *breakpoint*.
- Uma *strip*  $\pi[i..j]$  é uma trecho maximal em  $\pi$  tal que todos os pares  $(\pi_k, \pi_{k+1})$  são adjacências, para  $i \leq k < j$ .
- O número de *breakpoints* numa permutação  $\pi$  é denotado por  $b_t(\pi)$ .
- A única permutação sem *breakpoints* é a permutação identidade.
- Seja  $\Delta_{b_t}(\pi, \rho) = b_t(\pi \cdot \rho) - b_t(\pi)$ . Logo, temos que  $\Delta_{b_t}(\pi, \rho) \in \{-3, -2, -1, 0, 1, 2, 3\}$ .
- Podemos obter o seguinte limite inferior para o valor distância de transposição ( $d_t(\pi)$ ):

$$d_t(\pi) \geq \frac{b_t(\pi)}{3}$$

Algoritmo Ingênuo Baseado no Selection Sort para Ordenação por Transposições

Algoritmo 5: Selection Sort using Transpositions of Strips

Input:  $\pi, n$

$t \leftarrow 0$

while  $\pi \neq \iota$  do

$i \leftarrow 1$

  while  $\pi_i - \pi_{i-1} = 1$  do  $i \leftarrow i + 1$

$j \leftarrow i + 1$

  while  $\pi_j - \pi_{i-1} \neq 1$  do  $j \leftarrow j + 1$

$k \leftarrow j + 1$

  while  $\pi_k - \pi_{k-1} = 1$  do  $k \leftarrow k + 1$

$t \leftarrow t + 1$

$\pi \leftarrow \pi \cdot \rho(i, j, k)$

end

return  $t$

Algoritmo Ingênuo Baseado no Selection Sort para Ordenação por Transposições

- Complexidade:  $O(n^2)$ .
- Aproximação:
  - ▶ O algoritmo ingênuo remove pelo menos um *breakpoint* por transposição.
  - ▶ O algoritmo ingênuo garante o seguinte limite superior para distância de transposição:

$$d_t(\pi) \leq b_t(\pi) - 2$$

- ▶ Logo, o algoritmo ingênuo é um algoritmo de aproximação com fator:

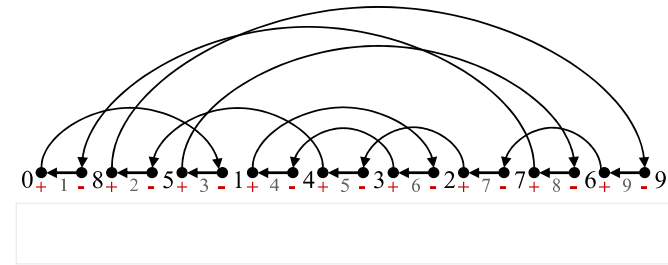
$$\frac{b_t(\pi) - 2}{\frac{b_t(\pi)}{3}} < 3$$



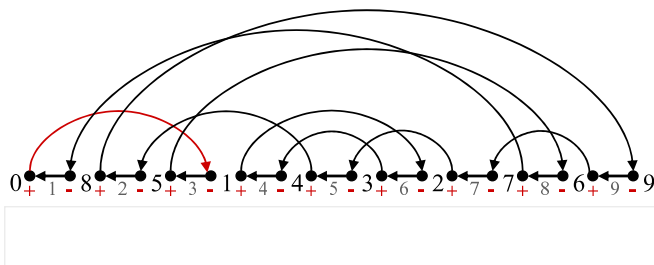
## Grafo de Ciclos Alternados

### Grafo de Ciclos Alternados

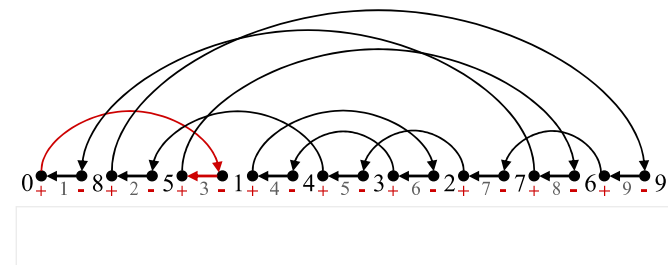
- Dada uma permutação  $\pi$  de  $n$  elementos defina o grafo de ciclos alternados como  $G(\pi) = (V, E_{black} \cup E_{gray})$ .
- Defina o conjunto de vértices como:
  - ▶  $V = \{+0, -\pi_1, +\pi_1, -\pi_2, +\pi_2, \dots, -\pi_n, +\pi_n, -(n+1)\}$ .
- Defina o conjunto de arestas pretas como:
  - ▶  $E_{black} = \{(-\pi_i, +\pi_{i-1}) | 1 \leq i \leq n+1\}$ .
- Defina o conjunto de arestas cinzas como:
  - ▶  $E_{gray} = \{+(i-1), -i | 1 \leq i \leq n+1\}$ .



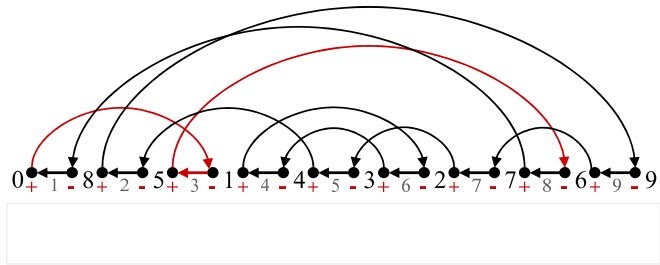
### Grafo de Ciclos Alternados



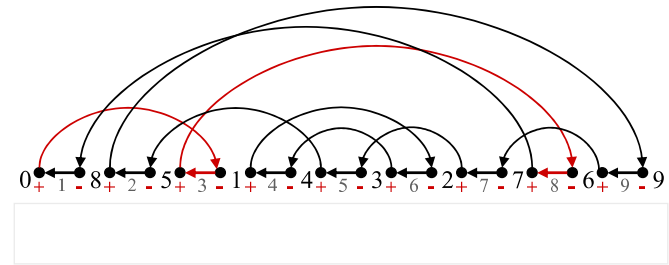
### Grafo de Ciclos Alternados



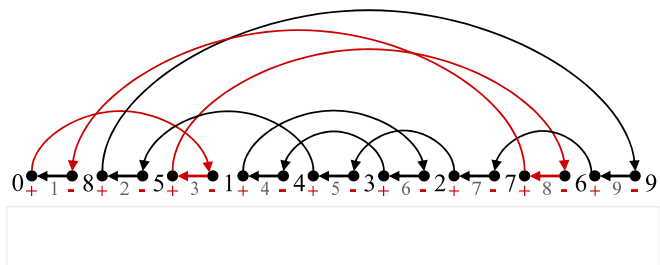
### Grafo de Ciclos Alternados



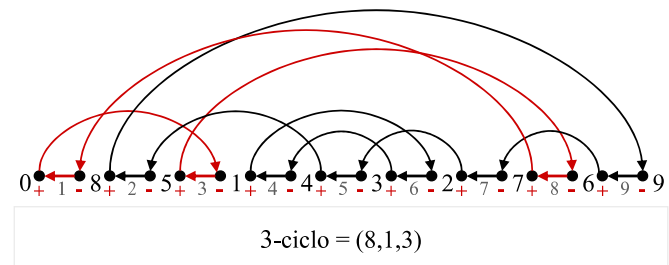
### Grafo de Ciclos Alternados



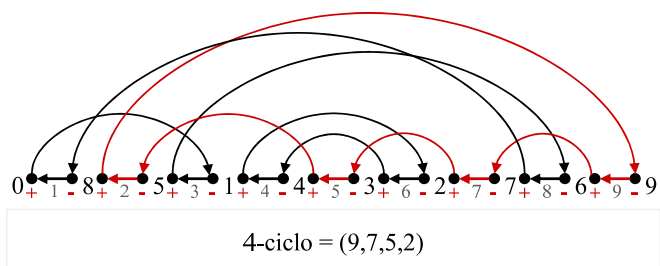
### Grafo de Ciclos Alternados



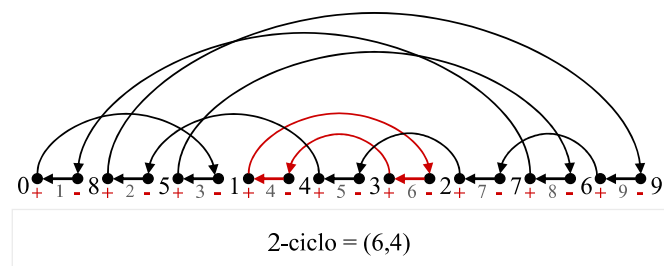
### Grafo de Ciclos Alternados



## Grafo de Ciclos Alternados



## Grafo de Ciclos Alternados

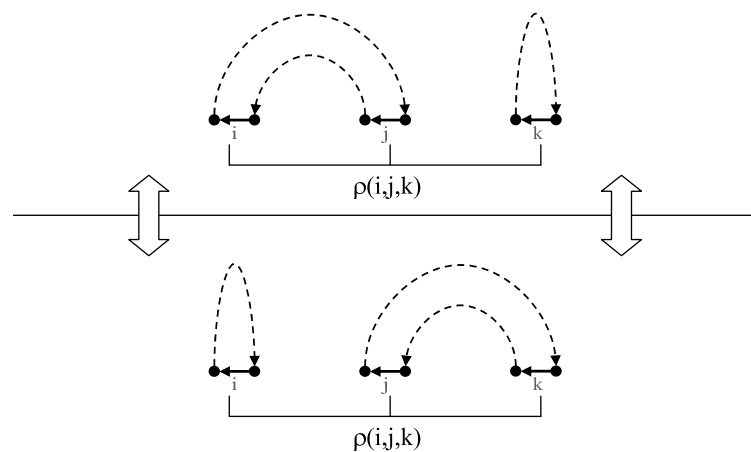


## Grafo de Ciclos Alternados

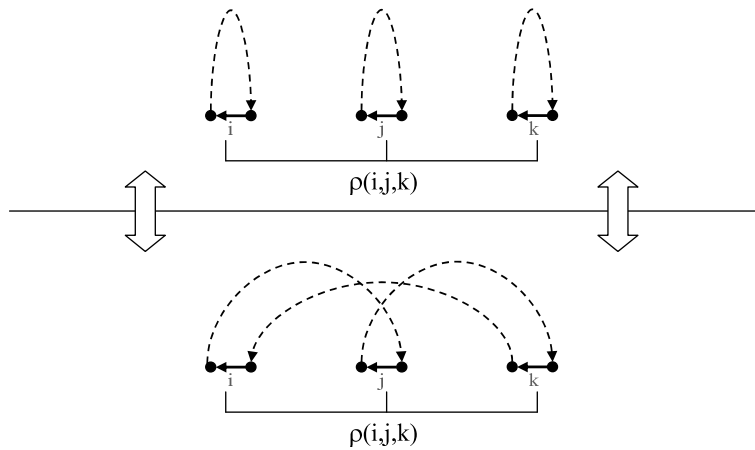
- Seja  $c(\pi)$  o número de ciclos alternados de  $G(\pi)$ .
- Seja  $c_{\text{odd}}(\pi)$  o número de ciclos alternados com número ímpar de arestas pretas de  $G(\pi)$ .
- A permutação identidade é a única cujo grafo de ciclos alternados possui  $n + 1$  ciclos, sendo todos ímpares ( $c(\iota) = c_{\text{odd}}(\iota) = n + 1$ ).
- Seja  $\Delta_c(\pi, \rho) = c(\pi \cdot \rho) - c(\pi)$ . Logo,  $\Delta_c(\pi, \rho) \in \{-2, 0, 2\}$ .
- Seja  $\Delta_{c_{\text{odd}}}(\pi, \rho) = c_{\text{odd}}(\pi \cdot \rho) - c_{\text{odd}}(\pi)$ . Pela paridade dos tamanhos dos ciclos, temos que  $\Delta_{c_{\text{odd}}}(\pi, \rho) \in \{-2, 0, +2\}$ .
- Usando a variação do número de ciclos ímpares podemos obter o seguinte limite inferior para o valor distância de transposição:

$$d_t(\pi) \geq \frac{n + 1 - c_{\text{odd}}(\pi)}{2}$$

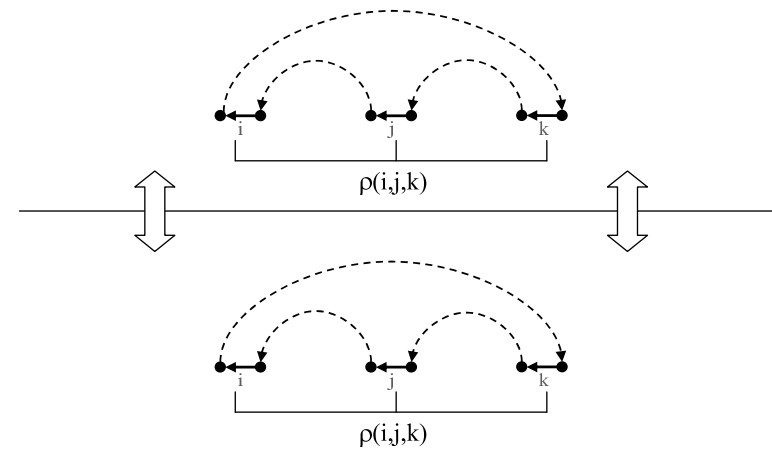
## Transposição Agindo em Ciclos Alternados



## Transposição Agindo em Ciclos Alternados



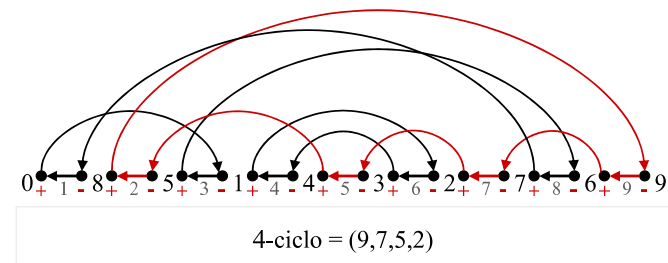
## Transposição Agindo em Ciclos Alternados



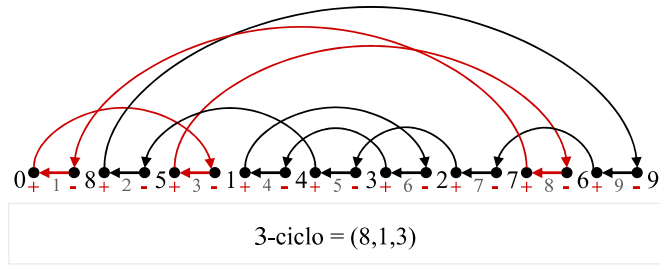
## Grafo de Ciclos Alternados

- As arestas pretas do grafo  $G(\pi)$  são numeradas de 1 a  $n + 1$ , sendo que a aresta  $(-\pi_i, +\pi_{i-1})$  recebe o rótulo  $i$ .
- Uma transposição  $\rho(i, j, k)$  age nas arestas pretas  $i, j$  e  $k$ .
- Uma transposição  $\rho$  em relação a  $\pi$  é chamada de  $x$ -move se e somente se  $\Delta_c(\pi, \rho) = x$ .
- Um ciclo alternado (ciclo de arestas de cores alternadas) é chamado  $k$ -ciclo se ele possuir  $2k$  arestas ( $k$  pretas e  $k$  cinzas).
- Um  $k$ -ciclo é identificado por suas arestas pretas, a partir da arestas de maior rótulo, de acordo com a ordem imposta pela orientação de suas arestas.
- Um  $k$ -ciclo  $C = (i_1, i_2, \dots, i_k)$  é chamado não orientado se  $i_1, i_2, \dots, i_k$  for uma sequência decrescente. Caso contrário, o ciclo é chamado de orientado.
- Uma transposição  $\rho(i, j, k)$  age num ciclo  $C$  se  $i, j, k \in C$ .

## Ciclo Não Orientado



## Ciclo Orientado



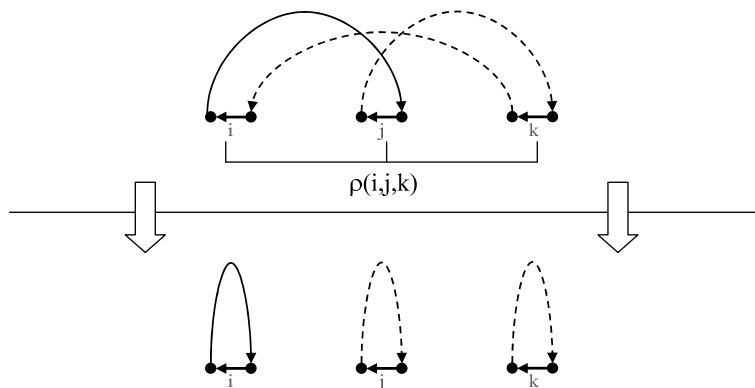
## 2-Move agindo num Ciclo Orientado

### Lema

Se  $C$  é um ciclo orientado, então existe um 2-move que age em  $C$ .

*Prova.* Seja  $C = (i_1, i_2, \dots, i_k)$  um ciclo orientado e seja  $3 \leq t \leq k$  um índice tal que  $i_t > i_{t-1}$ . Considere uma transposição  $\rho(i_{t-1}, i_t, i_1)$  agindo em  $C$ . Essa transposição cria um novo 1-ciclo (com os vértices  $+\pi_{i_{t-1}-1}$  e  $-\pi_{i_t}$ ), além de outros ciclos. Logo,  $\rho(i_{t-1}, i_t, i_1)$  é um 2-move.  $\square$

## 2-Move agindo num Ciclo Orientado



## Grafo de Ciclos Alternados

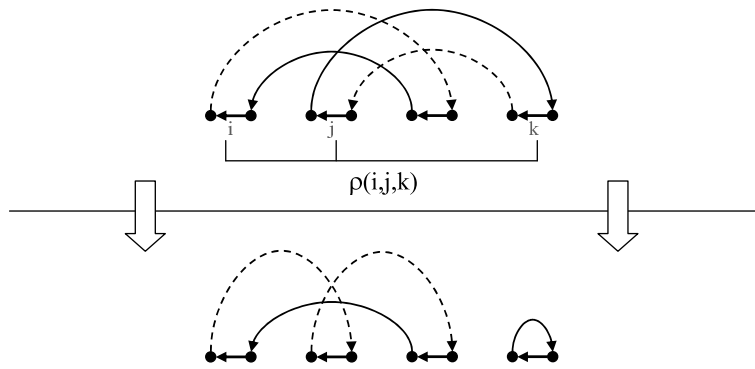
### Lema

Seja  $\pi \neq \iota$  uma permutação tal que  $G(\pi)$  não possua nenhum ciclo orientado. Logo, existe um good 0-move, ou seja, um 0-move que cria um ciclo orientado.

### Teorema

Qualquer permutação  $\pi \neq \iota$  pode ser ordenada com  $n + 1 - c(\pi)$  transposições.

### 0-Move que Cria um Ciclo Orientado



### Algoritmo de Aproximação para Ordenação por Transposições

---

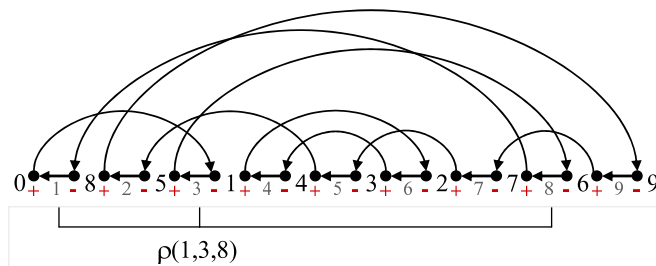
#### Algoritmo 6: Sorting by Transpositions

---

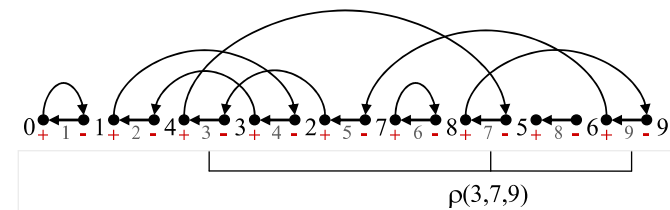
**Input:**  $\pi, n$   
 $t \leftarrow 0$   
**while**  $\pi \neq \iota$  **do**  
    **if**  $G(\pi)$  has an oriented cycle  $C$  **then**  
         $\rho \leftarrow$  a 2-move acting on  $C$   
    **end**  
    **else**  
         $\rho \leftarrow$  a good 0-move  
    **end**  
     $t \leftarrow t + 1$   
     $\pi \leftarrow \pi \cdot \rho$   
**end**  
**return**  $t$

---

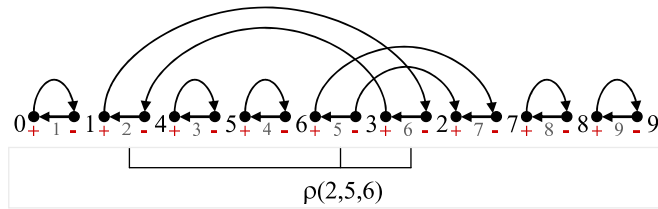
### Algoritmo de Aproximação para Ordenação por Transposições



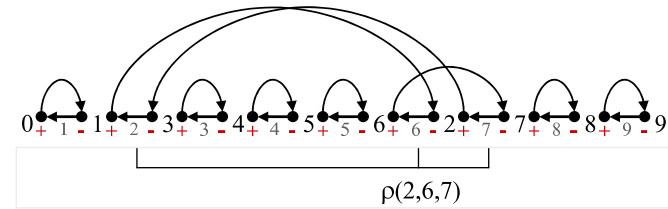
### Algoritmo de Aproximação para Ordenação por Transposições



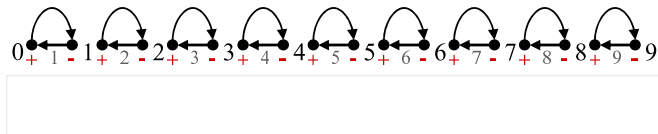
### Algoritmo de Aproximação para Ordenação por Transposições



### Algoritmo de Aproximação para Ordenação por Transposições



### Algoritmo de Aproximação para Ordenação por Transposições



### Algoritmo de Aproximação para Ordenação por Transposições

- Complexidade:  $O(n^2)$ .
- Aproximação:

$$\frac{n+1-c(\pi)}{\frac{n+1-c(\pi)}{2}} = 2$$

- Vineet Bafna e Pavel Pevzner, em 1998, apresentaram algoritmos de aproximação com fatores 3, 2, 1.75 e 1.5 para o problema de ordenação por transposições.
- Isaac Elias and Tzvika Hartman, em 2005, apresentaram um algoritmo de aproximação com fator 1.375 para o problema de ordenação por transposições. A prova da corretude do algoritmo é baseada em mais de 80 mil casos, que foram verificados computacionalmente.
- Laurent Bulteau, Guillaume Fertin e Irena Rusu provaram em 2011 que o problema de ordenação por transposições é  $\mathcal{NP}$ -Completo.

## Exercícios

### Exercício

Ordene, usando o algoritmo de 2-aproximação, as seguintes permutações:

- $\pi = (7, 1, 6, 2, 5, 3, 4)$
- $\pi = (7, 6, 5, 4, 3, 2, 1)$
- $\pi = (8, 7, 6, 5, 4, 3, 2, 1)$

## Calculando Todas as Distâncias de Rearranjo para Permutações Pequenas

- Construa um grafo onde cada vértice represente uma permutação de  $S_n$  (conjunto de todas as permutações de tamanho  $n$ ).
- Para cada vértice  $\pi$ , adicione arestas para todos os vértices  $\sigma$ , tal que  $\pi \cdot \rho = \sigma$ , para todos eventos de rearranjo  $\rho$  aplicáveis a  $\pi$ .
- Para computar as distâncias, faça uma busca em largura a partir de  $\iota$  para todos os demais vértices  $\pi$  do grafo, computando o número de vértices no caminho entre  $\iota$  e  $\pi$ . Neste caso estamos supondo que se  $\pi \cdot \rho = \sigma$  então existe um evento de rearranjo  $\rho'$  tal que  $\sigma \cdot \rho' = \pi$ .
- Este algoritmo tem complexidade polinomial no número de permutações de  $S_n$ , no entanto  $|S_n| = n!$  para permutações sem sinais e  $|S_n| = 2^n n!$  para permutações com sinais.
- Rearrangement Distance Database:  
<http://mirza.ic.unicamp.br:8080/bioinfo>