The problem
Fast Homing
Slow Homing
Counting bad cases

Sorting by placement and shift

Sergi Elizalde       Peter Winkler

Dartmouth College

SODA 2009
The *homing* algorithm

Given a permutation $\pi \in S_n$, repeat the following placement step:

- Choose an entry $\pi(i)$ such that $\pi(i) \neq i$.
- Place $\pi(i)$ in the correct position.
- Shift the other entries as necessary.
Main questions

▸ Does the algorithm always finish?
Main questions

- Does the algorithm always finish? YES
Main questions

- Does the algorithm always finish? **YES**
- How many steps does it take in the worst case...
Main questions

- Does the algorithm always finish?  YES
- How many steps does it take in the worst case... 
  - with a good choice of placements?
  - with a random choice of placements?
  - with a bad choice of placements?
The problem
Fast Homing
Slow Homing
Counting bad cases

Fast Homing: Well-chosen placements

Theorem

An algorithm that always places the smallest or largest available number will terminate in at most $n-1$ steps.
Fast Homing: Well-chosen placements

Theorem
An algorithm that always places the smallest or largest available number will terminate in at most $n-1$ steps.

Theorem
Let $k$ be the length of the longest increasing subsequence in $\pi$. Then no sequence of fewer than $n-k$ placements can sort $\pi$. 
Random placements

Theorem

The expected number of steps required by random homing from \( \pi \in S_n \) is at most \( \frac{n^2+n-2}{4} \).
Random placements

**Theorem**

The expected number of steps required by random homing from \( \pi \in S_n \) is at most \( \frac{n^2 + n - 2}{4} \).

**Proof.**

- Suppose that we have a permutation where \( k \) of the extremal numbers are home:

  123746589
Theorem

The expected number of steps required by random homing from \( \pi \in S_n \) is at most \( \frac{n^2 + n - 2}{4} \).

Proof.

- Suppose that we have a permutation where \( k \) of the extremal numbers are home:

  \[
  123746589
  \]

- With probability \( \leq \frac{2}{n-k} \), the next step will place an additional extremal number.
Random placements

Theorem

The expected number of steps required by random homing from \( \pi \in S_n \) is at most \( \frac{n^2 + n - 2}{4} \).

Proof.

- Suppose that we have a permutation where \( k \) of the extremal numbers are home:
  
  \[
  \begin{array}{cccccccc}
  1 & 2 & 3 & 7 & 4 & 6 & 5 & 8 & 9 \\
  \end{array}
  \]

- With probability \( \leq \frac{2}{n-k} \), the next step will place an additional extremal number.

- Total expected number of steps is \( \leq \sum_{k=0}^{n-2} \frac{n-k}{2} \). 

Starting from

\[2 \ 3 \ 4 \ 5 \ 6 \ 7 \ldots n \ 1\]

place always the leftmost possible entry:
Slow Homing: Example

Starting from

\[\begin{array}{cccccc}
2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}\ldots n & 1\]

place always the leftmost possible entry:
Slow Homing: Example

Starting from

\[ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ldots \ n \ 1 \]

place always the leftmost possible entry:

\[ 3 \ 2 \ 4 \ 5 \ 6 \ 7 \ldots \ n \ 1 \]
Starting from

2 3 4 5 6 7 \ldots n 1

place always the leftmost possible entry:

3 2 4 5 6 7 \ldots n 1
Slow Homing: Example

Starting from

\[234567\ldots n1\]

place always the leftmost possible entry:

\[324567\ldots n1\]
\[243567\ldots n1\]
Slow Homing: Example

Starting from

\[ 2 3 4 5 6 7 \ldots n 1 \]

place always the leftmost possible entry:

\[ 3 2 4 5 6 7 \ldots n 1 \]

\[ 2 4 3 5 6 7 \ldots n 1 \]
Slow Homing: Example

Starting from

\[ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ldots \ n \ 1 \]

place always the leftmost possible entry:

\[ 3 \ 2 \ 4 \ 5 \ 6 \ 7 \ldots \ n \ 1 \]
\[ 2 \ 4 \ 3 \ 5 \ 6 \ 7 \ldots \ n \ 1 \]
\[ 4 \ 2 \ 3 \ 5 \ 6 \ 7 \ldots \ n \ 1 \]
Slow Homing: Example

Starting from

\[2 \ 3 \ 4 \ 5 \ 6 \ 7 \ldots n \ 1\]

place always the leftmost possible entry:

\[3 \ 2 \ 4 \ 5 \ 6 \ 7 \ldots n \ 1\]
\[2 \ 4 \ 3 \ 5 \ 6 \ 7 \ldots n \ 1\]
\[4 \ 2 \ 3 \ 5 \ 6 \ 7 \ldots n \ 1\]
Slow Homing: Example

Starting from

\[2 \ 3 \ 4 \ 5 \ 6 \ 7 \ldots \ n \ 1\]

place always the leftmost possible entry:

\[3 \ 2 \ 4 \ 5 \ 6 \ 7 \ldots \ n \ 1\]
\[2 \ 4 \ 3 \ 5 \ 6 \ 7 \ldots \ n \ 1\]
\[4 \ 2 \ 3 \ 5 \ 6 \ 7 \ldots \ n \ 1\]
\[2 \ 3 \ 5 \ 4 \ 6 \ 7 \ldots \ n \ 1\]
Slow Homing: Example

Starting from

\[ 2 \, 3 \, 4 \, 5 \, 6 \, 7 \ldots \, n \, 1 \]

place always the leftmost possible entry:

\[ 3 \, 2 \, 4 \, 5 \, 6 \, 7 \ldots \, n \, 1 \]
\[ 2 \, 4 \, 3 \, 5 \, 6 \, 7 \ldots \, n \, 1 \]
\[ 4 \, 2 \, 3 \, 5 \, 6 \, 7 \ldots \, n \, 1 \]
\[ 2 \, 3 \, 5 \, 4 \, 6 \, 7 \ldots \, n \, 1 \]
Slow Homing: Example

Starting from

\[ \begin{array}{ccccccccc}
2 & 3 & 4 & 5 & 6 & 7 & \ldots & n & 1 \\
\end{array} \]

place always the leftmost possible entry:

\[ \begin{array}{ccccccccc}
3 & 2 & 4 & 5 & 6 & 7 & \ldots & n & 1 \\
2 & 4 & 3 & 5 & 6 & 7 & \ldots & n & 1 \\
4 & 2 & 3 & 5 & 6 & 7 & \ldots & n & 1 \\
2 & 3 & 5 & 4 & 6 & 7 & \ldots & n & 1 \\
3 & 2 & 5 & 4 & 6 & 7 & \ldots & n & 1 \\
\end{array} \]
Slow Homing: Example

Starting from

\[2\ 3\ 4\ 5\ 6\ 7\ldots n\ \ 1\]

place always the leftmost possible entry:

\[3\ 2\ 4\ 5\ 6\ 7\ldots n\ 1\]
\[2\ 4\ 3\ 5\ 6\ 7\ldots n\ 1\]
\[4\ 2\ 3\ 5\ 6\ 7\ldots n\ 1\]
\[2\ 3\ 5\ 4\ 6\ 7\ldots n\ 1\]
\[3\ 2\ 5\ 4\ 6\ 7\ldots n\ 1\]
Slow Homing: Example

Starting from

\[ \begin{array}{ccccc}
2 & 3 & 4 & 5 & 6 & 7 \ldots & n & 1 \\
\end{array} \]

place always the leftmost possible entry:

\[ \begin{array}{ccccc}
3 & 2 & 4 & 5 & 6 & 7 \ldots & n & 1 \\
2 & 4 & 3 & 5 & 6 & 7 \ldots & n & 1 \\
4 & 2 & 3 & 5 & 6 & 7 \ldots & n & 1 \\
2 & 3 & 5 & 4 & 6 & 7 \ldots & n & 1 \\
3 & 2 & 5 & 4 & 6 & 7 \ldots & n & 1 \\
2 & 5 & 3 & 4 & 6 & 7 \ldots & n & 1 \\
\end{array} \]
Slow Homing: Example

Starting from

$$2 \ 3 \ 4 \ 5 \ 6 \ 7 \ldots n \ 1$$

place always the leftmost possible entry:

$$3 \ 2 \ 4 \ 5 \ 6 \ 7 \ldots n \ 1$$
$$2 \ 4 \ 3 \ 5 \ 6 \ 7 \ldots n \ 1$$
$$4 \ 2 \ 3 \ 5 \ 6 \ 7 \ldots n \ 1$$
$$2 \ 3 \ 5 \ 4 \ 6 \ 7 \ldots n \ 1$$
$$3 \ 2 \ 5 \ 4 \ 6 \ 7 \ldots n \ 1$$
$$2 \ 5 \ 3 \ 4 \ 6 \ 7 \ldots n \ 1$$

It takes $$2^{n-1} - 1$$ steps to sort this permutation.
Main result

**Theorem**

*Homing always terminates in at most $2^{n-1} - 1$ steps.*
Theorem

Homing always terminates in at most \(2^{n-1} - 1\) steps.

To prove this, consider the reverse algorithm. We will show that, starting from the identity permutation, one can perform at most \(2^{n-1} - 1\) displacements.
Main result

Theorem

*Homing always terminates in at most* $2^{n-1} - 1$ *steps.*

To prove this, consider the reverse algorithm. We will show that, starting from the identity permutation, one can perform at most $2^{n-1} - 1$ displacements.

$$2^{n-1} - 1 = \underbrace{2^{n-2}} + \underbrace{2^{n-2} - 1}$$

until 1 and $n$ are displaced after displacing 1 and $n$
Lemma

After $2^{n-2}$ displacements, both 1 and $n$ have been displaced and will never be displaced again.
Lemma
After \(2^{n-2}\) displacements, both 1 and \(n\) have been displaced and will never be displaced again.

Proof.

- Note that 1 and \(n\) can each be displaced only once.
Lemma

After $2^{n-2}$ displacements, both $1$ and $n$ have been displaced and will never be displaced again.

Proof.

- Note that $1$ and $n$ can each be displaced only once.
- If after $2^{n-2}$ displacements one of these values hasn’t been displaced, then it played no role in the process.
Lemma

After $2^{n-2}$ displacements, both 1 and $n$ have been displaced and will never be displaced again.

Proof.

- Note that 1 and $n$ can each be displaced only once.
- If after $2^{n-2}$ displacements one of these values hasn’t been displaced, then it played no role in the process.
- Hence the remaining $n - 1$ numbers allowed more than $2^{n-2} - 1$ steps, contradicting the induction hypothesis.
The code of a permutation

We’ll show that after displacing 1 and \( n \), only \( 2^{n-2} - 1 \) more displacements can occur.
The code of a permutation

We’ll show that after displacing 1 and $n$, only $2^{n-2}-1$ more displacements can occur.

Assign to each permutation $\pi$ a code $\alpha(\pi) = \alpha_2\alpha_3 \ldots \alpha_{n-1}$, where

$$\alpha_i = \begin{cases} 
0 & \text{if entry } i \text{ is} \\
+ & \text{exactly to the right of} \\
- & \text{to the left of} \\
\text{home.}
\end{cases}$$
The code of a permutation

We’ll show that after displacing 1 and \( n \), only \( 2^{n-2} - 1 \) more displacements can occur.

Assign to each permutation \( \pi \) a code \( \alpha(\pi) = \alpha_2 \alpha_3 \ldots \alpha_{n-1} \), where

\[
\alpha_i = \begin{cases} 
0 & \text{if entry } i \text{ is exactly to the right of home.} \\
+ & \text{to the left of } \text{home.} \\
- & \text{home.}
\end{cases}
\]

Example

\[ \pi = 3 \ 5 \ 6 \ 1 \ 8 \ 4 \ 7 \ 2 \quad \longrightarrow \quad \alpha(\pi) = \]
The code of a permutation

We’ll show that after displacing 1 and \( n \), only \( 2^{n-2} - 1 \) more displacements can occur.

Assign to each permutation \( \pi \) a code \( \alpha(\pi) = \alpha_2 \alpha_3 \ldots \alpha_{n-1} \), where

\[
\alpha_i = \begin{cases} 
0 & \text{if entry } i \text{ is exactly to the right of home.} \\
+ & \text{if entry } i \text{ is to the left of home.}
\end{cases}
\]

Example

\( \pi = 3 5 6 1 8 4 7 2 \quad \rightarrow \quad \alpha(\pi) = + \)
The code of a permutation

We’ll show that after displacing 1 and $n$, only $2^{n-2}-1$ more displacements can occur.

Assign to each permutation $\pi$ a code $\alpha(\pi) = \alpha_2\alpha_3 \ldots \alpha_{n-1}$, where

$$\alpha_i = \begin{cases} 0 & \text{if entry } i \text{ is exactly to the right of home.} \\ + & \text{to the right of home.} \\ - & \text{to the left of home.} \end{cases}$$

Example

$$\pi = 3 \ 5 \ 6 \ 1 \ 8 \ 4 \ 7 \ 2 \quad \longrightarrow \quad \alpha(\pi) = + -$$
The code of a permutation

We’ll show that after displacing 1 and \( n \), only \( 2^{n-2} - 1 \) more displacements can occur.

Assign to each permutation \( \pi \) a code \( \alpha(\pi) = \alpha_2 \alpha_3 \ldots \alpha_{n-1} \), where

\[
\alpha_i = \begin{cases} 
0 & \text{if entry } i \text{ is exactly to the right of home.} \\
+ & \text{if entry } i \text{ is to the left of home.} \\
- & \text{if entry } i \text{ is to the left of home.}
\end{cases}
\]

Example

\( \pi = 3 5 6 1 8 4 7 2 \) \( \longrightarrow \) \( \alpha(\pi) = + - + \)
The code of a permutation

We’ll show that after displacing 1 and $n$, only $2^{n-2}-1$ more displacements can occur.

Assign to each permutation $\pi$ a code $\alpha(\pi) = \alpha_2\alpha_3 \ldots \alpha_{n-1}$, where

$$\alpha_i = \begin{cases} 
0 & \text{if entry } i \text{ is exactly to the right of home.} \\
+ & \text{to the right of} \\
- & \text{to the left of} 
\end{cases}$$

Example

$\pi = 3\ 5\ 6\ 1\ 8\ 4\ 7\ 2 \quad \rightarrow \quad \alpha(\pi) = +\ -\ -\ +\ -$
The code of a permutation

We’ll show that after displacing 1 and $n$, only $2^{n-2} - 1$ more displacements can occur.

Assign to each permutation $\pi$ a code $\alpha(\pi) = \alpha_2 \alpha_3 \ldots \alpha_{n-1}$, where

$$\alpha_i = \begin{cases} 0 & \text{if entry } i \text{ is exactly to the right of home.} \\ + & \text{if entry } i \text{ is to the left of } \\ - & \text{to the left of } \end{cases}$$

Example

$$\pi = 3 \ 5 \ 6 \ 1 \ 8 \ 4 \ 7 \ 2 \quad \longrightarrow \quad \alpha(\pi) = + \ - \ + \ - \ - \ -$$
We’ll show that after displacing 1 and $n$, only $2^{n-2}-1$ more displacements can occur.

Assign to each permutation $\pi$ a code $\alpha(\pi) = \alpha_2\alpha_3 \ldots \alpha_{n-1}$, where

$$\alpha_i = \begin{cases} 0 & \text{if entry } i \text{ is} \\ + & \text{exactly to the right of} \\ - & \text{to the left of} \end{cases}$$

home.

Example

$$\pi = 3 5 6 1 8 4 7 2 \rightarrow \alpha(\pi) = + - + - - - 0$$
The weight of a code

\[ \alpha = + - + - - 0 \]

Define the weight of a code \( \alpha \) recursively:
The weight of a code

\[ \alpha = + - + - - 0 \]
\[ \begin{array}{cccccc}
5 & 1 & 3 & 3 & 4 \\
\end{array} \]

Define the weight of a code \( \alpha \) recursively:

- For each \(-\), count the number of symbols to its left, and for each \(+\), count the number of symbols to its right.
The weight of a code

\[ \alpha = + - + - - 0 \]
\[
\begin{array}{cccccc}
5 & 1 & 3 & 3 & 4 \\
\end{array}
\]
\[ \hat{\alpha} = - + - - 0 \]

Define the weight of a code \( \alpha \) recursively:

- For each \(-\), count the number of symbols to its left, and for each \(+\), count the number of symbols to its right.
- Let \( d \) be the largest of these numbers, and let \( \hat{\alpha} \) be the code obtained by deleting the corresponding symbol.
The weight of a code

\[ \alpha = + - + - - 0 \]
\[ 5 \quad 1 \quad 3 \quad 3 \quad 4 \]
\[ \hat{\alpha} = - + - - 0 \]

Define the weight of a code \( \alpha \) recursively:

- For each \(-\), count the number of symbols to its left, and for each \(+\), count the number of symbols to its right.
- Let \( d \) be the largest of these numbers, and let \( \hat{\alpha} \) be the code obtained by deleting the corresponding symbol.
- Define

\[
w(\alpha) = 2^d + w(\hat{\alpha}).
\]
The weight of a code: example

\[ w( + - + - - 0 ) \]
The weight of a code: example

\[ w( + - + - - 0 ) = 5 1 3 3 4 \]
The weight of a code: example

\[ w( + - + - - 0 ) \]
\[ 5 1 3 3 4 \]
\[ = 2^5 + w( - + - - 0 ) \]
The weight of a code: example

\[ w( + - + - - 0 ) \]
\[ 5 \ 1 \ 3 \ 3 \ 4 \]
\[ = 2^5 + w( - + - - 0 ) \]
\[ 0 \ 3 \ 2 \ 3 \]
The weight of a code: example

\[
\begin{align*}
  w\left( + \; - \; + \; - \; - \; 0 \right) &= 2^5 + w\left( - \; + \; - \; - \; - \; 0 \right) \\
  &= 2^5 + 2^3 + w\left( - \; + \; - \; 0 \right)
\end{align*}
\]
The weight of a code: example

\[ w( + - + - - 0 ) \]
\[ = 2^5 + w( - + - - 0 ) \]
\[ = 2^5 + 2^3 + w( - + - 0 ) \]
\[ = 2^5 + 2^3 + w( - + - 0 ) \]
The weight of a code: example

\[ w( + - + - - 0 ) \]
\[ = 2^5 + w( - + - - 0 ) \]
\[ = 2^5 + 2^3 + w( - + - 0 ) \]
\[ = 2^5 + 2^3 + 2^2 + w( - + 0 ) \]
The weight of a code: example

\[
\begin{align*}
w( + & - + - - 0 ) \\
5 & 1 3 3 4 \\
= 2^5 + w( - & + - - 0 ) \\
0 & 3 2 3 \\
= 2^5 + 2^3 + w( - & + - 0 ) \\
0 & 2 2 \\
= 2^5 + 2^3 + 2^2 + w( - & + 0 ) \\
0 & 1
\end{align*}
\]
The weight of a code: example

\[
\begin{align*}
w( + & - + - - 0 ) \\
5 & 1 3 3 4 \\
= 2^5 + w( - + - - 0 ) \\
0 & 3 2 3 \\
= 2^5 + 2^3 + w( - + - 0 ) \\
0 & 2 2 \\
= 2^5 + 2^3 + 2^2 + w( - + 0 ) \\
0 & 1 \\
= 2^5 + 2^3 + 2^2 + 2^1 + w( - 0 )
\end{align*}
\]
The weight of a code: example

\[ w( + - + - - 0 ) \]
\[
\begin{array}{cccc}
5 & 1 & 3 & 3 \\
5 & 1 & 3 & 3 \\
\end{array}
\]

\[ = 2^5 + w( - + - - 0 ) \]
\[
\begin{array}{cccc}
0 & 3 & 2 & 3 \\
0 & 3 & 2 & 3 \\
\end{array}
\]

\[ = 2^5 + 2^3 + w( - + - 0 ) \]
\[
\begin{array}{ccc}
0 & 2 & 2 \\
0 & 2 & 2 \\
\end{array}
\]

\[ = 2^5 + 2^3 + 2^2 + w( - + 0 ) \]
\[
\begin{array}{cc}
0 & 1 \\
0 & 1 \\
\end{array}
\]

\[ = 2^5 + 2^3 + 2^2 + 2^1 + w( - 0 ) \]
\[
\begin{array}{c}
0 \\
0 \\
\end{array}
\]
The weight of a code: example

\[w( + - + - - 0 )
\]

\[= 2^5 + w( - + - - 0 )
\]

\[= 2^5 + 2^3 + w( - + - 0 )
\]

\[= 2^5 + 2^3 + 2^2 + w( - + 0 )
\]

\[= 2^5 + 2^3 + 2^2 + 2^1 + w( - 0 )
\]

\[= 2^5 + 2^3 + 2^2 + 2^1 + 2^0 + w( 0 )
\]
The weight of a code: example

\[
\begin{align*}
w( & + - + - - 0 ) \\
5 & 1 3 3 4 \\
= 2^5 & + w( - + - - 0 ) \\
0 & 3 2 3 \\
= 2^5 & + 2^3 + w( - + - 0 ) \\
0 & 2 2 \\
= 2^5 & + 2^3 + 2^2 + w( - + 0 ) \\
0 & 1 \\
= 2^5 & + 2^3 + 2^2 + 2^1 + w( - 0 ) \\
0 & \\
= 2^5 & + 2^3 + 2^2 + 2^1 + 2^0 + w( 0 ) \\
= 2^5 & + 2^3 + 2^2 + 2^1 + 2^0 = 47
\end{align*}
\]
Bound on the weight

Lemma

The maximum of $w(\alpha)$ over codes $\alpha$ of length $k$ is $2^k - 1$, for codes of the form $++\cdots++--\cdots--$. 
Lemma

The maximum of $w(\alpha)$ over codes $\alpha$ of length $k$ is $2^k - 1$, for codes of the form $++\cdots+−−\cdots−$.

Proof.

In the recursion,

$$w(\alpha) \leq 2^{k-1} + w(\hat{\alpha}),$$

with equality when a $−$ is deleted from the right or a $+$ from the left.
The weight increases at each displacement

**Lemma**

Let $\pi \in S_n$ with $\pi(1) \neq 1$ and $\pi(n) \neq n$, and let $\pi'$ be the result of applying some displacement to $\pi$. Let $\alpha = \alpha(\pi)$ and $\alpha' = \alpha(\pi')$. Then

$$w(\alpha') > w(\alpha).$$
Lemma
Let \( \pi \in S_n \) with \( \pi(1) \neq 1 \) and \( \pi(n) \neq n \), and let \( \pi' \) be the result of applying some displacement to \( \pi \). Let \( \alpha = \alpha(\pi) \) and \( \alpha' = \alpha(\pi') \). Then
\[
  w(\alpha') > w(\alpha).
\]

Proof sketch.

- A number \( i \) can be displaced iff \( \alpha_i = 0 \) in the code.
The weight increases at each displacement

Lemma
Let $\pi \in S_n$ with $\pi(1) \neq 1$ and $\pi(n) \neq n$, and let $\pi'$ be the result of applying some displacement to $\pi$. Let $\alpha = \alpha(\pi)$ and $\alpha' = \alpha(\pi')$. Then

$$w(\alpha') > w(\alpha).$$

Proof sketch.

- A number $i$ can be displaced iff $\alpha_i = 0$ in the code.
- If it is displaced to the left, then $\alpha_i$ becomes a $-\$, and some of the other entries can change from $-$ to 0 or from 0 to $+$. 

Sergi Elizalde, Peter Winkler

Sorting by placement and shift
The weight increases at each displacement

Lemma

Let \( \pi \in S_n \) with \( \pi(1) \neq 1 \) and \( \pi(n) \neq n \), and let \( \pi' \) be the result of applying some displacement to \( \pi \). Let \( \alpha = \alpha(\pi) \) and \( \alpha' = \alpha(\pi') \). Then

\[
w(\alpha') > w(\alpha).
\]

Proof sketch.

- A number \( i \) can be displaced iff \( \alpha_i = 0 \) in the code.
- If it is displaced to the left, then \( \alpha_i \) becomes a \( - \), and some of the other entries can change from \( - \) to \( 0 \) or from \( 0 \) to \( + \).
- It can be shown that this increases the weight of the code.
The weight increases at each displacement

Lemma
Let $\pi \in S_n$ with $\pi(1) \neq 1$ and $\pi(n) \neq n$, and let $\pi'$ be the result of applying some displacement to $\pi$. Let $\alpha = \alpha(\pi)$ and $\alpha' = \alpha(\pi')$. Then

$$w(\alpha') > w(\alpha).$$

Proof sketch.

- A number $i$ can be displaced iff $\alpha_i = 0$ in the code.
- If it is displaced to the left, then $\alpha_i$ becomes a $-$, and some of the other entries can change from $-$ to $0$ or from $0$ to $+$.  
- It can be shown that this increases the weight of the code.
- Similarly if $i$ is displaced to the right.
Finishing the proof

Combining these lemmas, the maximum number of displacements is

- at most $2^{n-2}$ until 1 and $n$ are displaced, plus
- at most $2^{n-2} - 1$ after 1 and $n$ have been displaced.
Finishing the proof

Combining these lemmas, the maximum number of displacements is

- at most $2^{n-2}$ until 1 and $n$ are displaced, plus
- at most $2^{n-2} - 1$ after 1 and $n$ have been displaced.

So at most $2^{n-1} - 1$ in total.
The number of worst-case permutations

\[ h(\pi) = \text{max. length of a seq. of placements from } \pi \text{ to } 12 \ldots n. \]
The number of worst-case permutations

\[ h(\pi) = \max \text{ length of a seq. of placements from } \pi \text{ to } 12 \ldots n. \]

\[ M_n = \{ \pi \in S_n : h(\pi) = 2^{n-1} - 1 \}. \]
The number of worst-case permutations

\[ h(\pi) = \text{max. length of a seq. of placements from } \pi \text{ to } 12 \ldots n. \]

\[ M_n = \{ \pi \in S_n : h(\pi) = 2^{n-1} - 1 \}. \]

Ex: \( 23 \ldots n1 \in M_n \).
The number of worst-case permutations

\[ h(\pi) = \text{max. length of a seq. of placements from } \pi \text{ to } 12 \ldots n. \]

\[ M_n = \{ \pi \in S_n : h(\pi) = 2^{n-1} - 1 \}. \]

Ex: \( 23 \ldots n1 \in M_n. \)

Theorem

\[ B_{n-1} \leq |M_n| \leq (n - 1)! , \]

where \( B_n = n\text{-th Bell number} = \# \text{ partitions of } \{1, 2, \ldots, n\}. \)
The number of worst-case permutations

\[ h(\pi) = \max \text{ length of a seq. of placements from } \pi \text{ to } 12\ldots n. \]

\[ M_n = \{ \pi \in S_n : h(\pi) = 2^{n-1} - 1 \}. \]

Ex: 23\ldots n1 \in M_n.

**Theorem**

\[ B_{n-1} \leq |M_n| \leq (n - 1)!, \]

where \( B_n = n\text{-th Bell number} = \# \text{ partitions of } \{1, 2, \ldots, n\}. \)

\( B_n \) grows super-exponentially.
The number of worst-case permutations

\[ f_{i,j} = |\{ \pi \in M_{i+j} : \alpha(\pi) = \underbrace{++ \cdots +}_{i-1} \underbrace{- - \cdots -}_{j-1} \}| \]
The number of worst-case permutations

\[ f_{i,j} = |\{\pi \in M_{i+j} : \alpha(\pi) = \underbrace{+ + \cdots +}_{i-1} - \underbrace{- - \cdots -}_{j-1}\}| \]

\[ F(u, v) = \sum_{i,j \geq 1} f_{i,j} u^i v^j \]
The number of worst-case permutations

\[ f_{i,j} = |\{ \pi \in M_{i+j} : \alpha(\pi) = \overbrace{+ + \cdots +}^{i-1} - - \cdots - \}| \]

\[ F(u, v) = \sum_{i,j \geq 1} f_{i,j} u^i v^j \]

\[ |M_n| = \sum_{i+j=n} f_{i,j} \text{ is the coefficient of } t^n \text{ in } F(t, t). \]
The number of worst-case permutations

\[ f_{i,j} = \left| \{ \pi \in M_{i+j} : \alpha(\pi) = \underbrace{++ \cdots +}_{i-1} \underbrace{- - \cdots -}_{j-1} \} \right| \]

\[ F(u, v) = \sum_{i,j \geq 1} f_{i,j} u^i v^j \]

\[ |M_n| = \sum_{i+j=n} f_{i,j} \] is the coefficient of \( t^n \) in \( F(t, t) \).

Theorem

\[ F(u, v) = uv + uv \frac{\partial}{\partial u} F(u, v) + uv \frac{\partial}{\partial v} F(u, v) - u^2 v^2 \frac{\partial^2}{\partial u \partial v} F(u, v) \]
The problem
Fast Homing
Slow Homing
Counting bad cases

23451 34251 34521 45231 45321 45312 45132 43512 45123 45132 35412 45123 45124 51243

24351 24513 24531 24351 35142 51432 35142 53124 53134 32451 34251 51243

23514 35214 35421 54213 54231 52431 52341 53124 53142 45321 54312 45213 35412

32514 25431 54321 53142 53214 52431 52341 53124 53142 45321 54312 45213 35412

23415 34152 34512 33214 41523 41532 31452

23415 34152 34512 33214 41523 41532 31452

12345 12345 12345 12345 12345 12345 12345

Sergi Elizalde, Peter Winkler

Sorting by placement and shift
The problem
Fast Homing
Slow Homing
Counting bad cases

THANK YOU

Sergi Elizalde, Peter Winkler
Sorting by placement and shift