Pancakes With A Problem

Steven Rudich
The chef at our place is sloppy, and when he prepares a stack of pancakes they come out all different sizes. Therefore, when I deliver them to a customer, on the way to the table I rearrange them (so that the smallest winds up on top, and so on, down to the largest at the bottom) by grabbing several from the top and flipping them over, repeating this (varying the number I flip) as many times as necessary.

Developing A Notation: Turning pancakes into numbers
Developing A Notation:
Turning pancakes into numbers

5
2
3
4
1
Developing A Notation:
Turning pancakes into numbers

4 Flips Are Sufficient
Algebraic Representation

\[ X = \text{The smallest number of flips required to sort:} \]

? \leq X \leq 4

Algebraic Representation

\[ X = \text{The smallest number of flips required to sort:} \]

? \leq X \leq 4
4 Flips Are Necessary

Flip 1 has to put 5 on bottom.
Flip 2 must bring 4 to top.

? \leq X \leq 4

Lower Bound
4 \leq X \leq 4

Upper Bound

Lower Bound

X = 4

5^{th} Pancake Number

P_5 = \text{The number of flips required to sort the worst case stack of 5 pancakes.}

? \leq P_5 \leq ?
**5th Pancake Number**

\[ P_5 = \text{The number of flips required to sort the worst case stack of 5 pancakes.} \]

4 \leq P_5 \leq ?

**The 5th Pancake Number: The MAX of the X’s**

\[
\begin{align*}
X_1 & : 1 \\
X_2 & : 2 \\
X_3 & : 3 \\
X_{119} & : 1 \\
X_{120} & : 1
\end{align*}
\]

<table>
<thead>
<tr>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>X_{119}</th>
<th>X_{120}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
X_1 & : 1 \\
X_2 & : 2 \\
X_3 & : 3 \\
X_{119} & : 1 \\
X_{120} & : 1
\end{align*}
\]

\[
\begin{align*}
X_1 & : 1 \\
X_2 & : 2 \\
X_3 & : 3 \\
X_{119} & : 1 \\
X_{120} & : 1
\end{align*}
\]
**nth Pancake Number**

\[ P_n = \text{The number of flips required to sort the worst case stack of } n \text{ pancakes.} \]

\[ ? \leq P_n \leq ? \]

**Upper Bound**

**Lower Bound**

What bounds can you prove on \( P_n \)?
Bring To Top Method

Bring biggest to top. Place it on bottom. Bring next largest to top. Place second from bottom. And so on...

Upper Bound On $P_n$:
Bring To Top Method For $n$ Pancakes

If $n=1$, no work - we are done.
Otherwise, flip pancake $n$ to top and then flip it to position $n$.

Now use: Bring To Top Method For $n-1$ Pancakes

Total Cost: at most $2(n-1) = 2n - 2$ flips.
Better Upper Bound on $P_n$:
*Bring To Top Method For n Pancakes*

If $n=2$, at most one flip, we are done.
Otherwise, flip pancake $n$ to top and then flip it to position $n$.

Now use: *Bring To Top Method For n-1 Pancakes*

Total Cost: at most $2(n-2) + 1 = 2n - 3$ flips.

Bring to top not always optimal for a particular stack

5 flips, but can be done in 4 flips
What bounds can you prove on $P_n$?

**Breaking Apart Argument**

Suppose a stack $S$ contains a pair of adjacent pancakes that will not be adjacent in the sorted stack. Any sequence of flips that sorts stack $S$ must involve one flip that inserts the spatula between that pair and breaks them apart.
Breaking Apart Argument

Suppose a stack $S$ contains a pair of adjacent pancakes that will not be adjacent in the sorted stack. Any sequence of flips that sorts stack $S$ must involve one flip that inserts the spatula between that pair and breaks them apart. Furthermore, this same principle is true of the “pair” formed by the bottom pancake of $S$ and the plate.

$n \leq P_n$

Suppose $n$ is even. $S$ contains $n$ pairs that will need to be broken apart during any sequence that sorts stack $S$. 
Suppose $n$ is odd. $S$ contains $n$ pairs that will need to be broken apart during any sequence that sorts stack $S$.

Detail: This construction only works when $n > 3$.

$n \leq P_n \leq 2n - 3$

Bring To Top is within a factor of two of optimal.
n ≤ P_n ≤ 2n - 3

So starting from ANY stack we can get to the sorted stack using no more than P_n flips.

From ANY stack to Sorted Stack in · P_n

Can you see why we can get from the Sorted stack to ANY stack in · P_n flips?

Reverse the sequences we use to sort.
Can you see why we can get from ANY stack \( S \) to ANY stack \( T \) in \( \cdot P_n \) flips?

Rename the pancakes in \( S \) to be \( 1, 2, 3, \ldots, n \). Rewrite \( T \) using the new naming scheme that you used for \( S \). \( T \) will be some list: \( \pi(1), \pi(2), \ldots, \pi(n) \). The sequence of flips that brings the sorted stack to \( \pi(1), \pi(2), \ldots, \pi(n) \) will bring \( S \) to \( T \).

\[
\begin{align*}
S: & \quad 4, 3, 5, 1, 2 \\
& \quad 1, 2, 3, 4, 5 \\
T: & \quad 5, 2, 4, 3, 1 \\
& \quad 3, 5, 1, 2, 4
\end{align*}
\]
The Known Pancake Numbers

<table>
<thead>
<tr>
<th>n</th>
<th>( P_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
</tr>
</tbody>
</table>

\( P_{14} \) Is Unknown

14! Orderings of 14 pancakes.

\[ 14! = 87,178,291,200 \]
### General Versus Particular

<table>
<thead>
<tr>
<th>General</th>
<th>Particular</th>
</tr>
</thead>
<tbody>
<tr>
<td>We proved bounds for all pancake numbers</td>
<td>Brute force gave exact bounds for a small set of pancake numbers</td>
</tr>
</tbody>
</table>

### Is This Really Computer Science?

\[
\frac{17}{16}n \leq P_n \leq \frac{5n+5}{3}
\]

\[(15/14)n \leq P_n \leq (5n+5)/3\]


**Pancake Network:**
**Definition For n! Nodes**

For each node, assign it the name of one of the n! stacks of n pancakes.

Put a wire between two nodes if they are one flip apart.
Pancake Network: Message Routing Delay

What is the maximum distance between two nodes in the network?

Pancake Network: Reliability

If up to $n-2$ nodes get hit by lightning the network remains connected, even though each node is connected to only $n-1$ other nodes.

The Pancake Network is optimally reliable for its number of edges and nodes.
Mutation Distance

Head Cabbage
(\textit{Brassica oleracea capitata})

Turnip
(\textit{Brassica rapa})

©1997 The Learning Company, Inc.

One “Simple” Problem

A host of problems and applications at the frontiers of science
**Lower Bound On $P_n$**

*Adjacency in a Stack*: Any pair of consecutively sized pancakes next to each other in the stack.

*Adjacency Count of a stack*: number of Adjacencies in the stack PLUS 1 more if the largest pancake is at the bottom.
Examples Of Adjacency Counts

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Adjacency Count Zero
(for n > 3)

n even

2
4
6
8

\ldots

n
1
3
5
7

\ldots

n-1

n odd

1
3
5
7

\ldots

n
2
4
6
8

\ldots

n-1

n even

2
4
6
8

\ldots

n
1
3
5
7

\ldots

n-1
Adjacent Lemma:
Each flip can raise the Adjacency Count by at most one.

There is a stack of \( n \) pancakes of Adjacency Count 0. The sorted stack has count \( n \). Each flip increases the count by at most 1.

Hence it requires at least \( n \) flips to sort the original stack.