Genome Rearrangements

ARE YOU A MAN OR A MOUSE?
Revisiting Greedy Algorithms

An iterative algorithm where, at each step, one takes what seems to be the immediately best option

- **Cons**
  - It may return incorrect results
  - It may require more steps than necessary

- **Pros**
  - It often requires very little time to make a greedy choice
  - Choices are considered independently

Have we seen greedy algorithms in previous lectures?
The chef at "Breadman's" is sloppy.

He makes pancakes of nonuniform sizes, and just throws them onto the plate.

Before the waitress delivers them to your table, she rearranges them so that the smaller pancakes are attacked on the larger ones.

Since she has only one free hand to perform this culinary rearrangement, she does it by inserting a spatula into the stack and then she flips all the cakes above the spatula. She repeats the process over and over until the entire stack is sorted.

I was wondering, how many such flips are needed for her to transform a sloppy stack into a sorted one?
Pancake Flipping Problem: Formulation

- **Goal**: Given a stack of \( n \) pancakes, what is the minimum number of flips to rearrange them into a sorted stack (small on top of large)?
- **Input**: A permutation \( \Pi \)
- **Output**: A series of *prefix reversals*, \( \rho_1, \rho_2, \rho_t \) that transforms \( \Pi \) into the identity permutation such that \( t \) is minimal.

\[
\Pi = \pi_1, \ldots, \pi_{i-1}, \underline{\pi_i}, \pi_{i+1} \ldots \pi_n \\
\Pi = \pi_i, \pi_{i-1}, \ldots, \pi_1, \pi_{i+1} \ldots \pi_n
\]
Turning Pancakes into Numbers
"Bring-to-Top" Algorithm

1. Flip the biggest pancake to the top of the stack
2. Flip the entire stack \((n)\), to place the biggest pancake on the bottom
3. Flip the next largest pancake to the top
4. Flip the \(n-1\) top pancakes, thus, placing the 2nd largest on top of the largest
5. Repeat until sorted
"Bring-to-Top" Algorithm for $n$ Pancakes

- If $n = 1$, the smallest pancake is already on top - we are done
- Otherwise, flip pancake $n$ to the top, and then flip it to position $n$
- Now use:

\[
\text{Bring-to-Top}(n) \\
\text{Pancakes}
\]

- Analysis:
  - Performs 2 flips to put the $i$th pancake into its correct position.
  - There are $(n-1)$ pancakes to sort
  - So, no more then $2(n - 1)$ flips are needed to sort the entire stack
Is that the best we can do?

- Our algorithm is correct, but can it be done with fewer flips?
- Apply it to the following stack

![Diagram of a stack with numbers 1 to 5 undergoing flips](image)

- The "Bring-to-Top" algorithm did it in only 5 flips! The predicted "8" flips is an upper-bound on the number of flips required by "Bring-to-top"
- Is there another algorithm that can always rearrange the stack in fewer flips?
You can do it in 4 flips!

William Gates III (yeah, the Micorsoft guy) and Christos Papadimitriou showed in the mid-1970s that this problem can be solved by at least $17/16 \, n$ and at most $5/3 \, (n+1)$ prefix reversals (flips) of $n$ pancakes.
A Serious Scientific Problem

- Some organisms are obviously similar...
- Some organisms are obviously different...
- Some are close calls...
- The differences are all in their genes!
- And the gene order is important!
Humans and mice have similar genomes, but their genes are ordered differently
- \( \approx 245 \) rearrangements
- \( \approx 300 \) large synteny blocks
Genomes are Connected

- What are the synteny blocks, and how do we find them?
- What is the architecture of the common ancestral genome?
- What is the evolutionary scenario for transforming one genome into the other?
Rat Consortium, Nature 2004

Rearrangement Events:

- Reversals
- Fusions
- Fissions
- Translocations
Blocks represent conserved genes

Reversals, or inversions, are particularly relevant to speciation. Recombinations cannot occur between reversed and normally ordered segments.
Reversals

- Blocks represent conserved genes
- In the course of evolution, blocks 1 ... 10 could be reordered as 1 2 3 8 7 6 5 4 9 10
Reversal Breakpoints

- The inversion introduced two breakpoints (disruptions in order).
Other Types of Rearrangements

Translocation

Fusion

Fission
Reversals and Gene Orders

- Gene order can be represented by a permutation $\Pi$:

$$\Pi = \pi_1 \ldots \pi_{i-1} \pi_i \pi_{i+1} \ldots \pi_{j-1} \pi_j \pi_{j+1} \ldots \pi_n$$

- Reversal $\rho(i, j)$ reverses (flips) the elements from $i$ to $j$ in $\Pi$. 

Reversal Examples

\[ \Pi = 1, 2, 3, 4, 5, 6, 7, 8 \]
\[ \rho(3, 5) \downarrow \]
\[ \Pi = 1, 2, 5, 4, 3, 6, 7, 8 \]
\[ \rho(5, 6) \downarrow \]
\[ \Pi = 1, 2, 5, 4, 6, 3, 7, 8 \]
"Reversal Distance" Problem

- **Goal**: Given two permutations over $n$ elements, find the shortest series of reversals that transforms one into another.
- **Input**: Permutations $\Pi$ and $\Sigma$
- **Output**: A series of reversals $\rho_1(s_1, e_1), \rho_2(s_2, e_2), \ldots \rho_t(s_t, e_t)$ transforming $\Pi$ to $\Sigma$ such that $t$ is minimal.
- $t$ - reversal distance between $\Pi$ and $\Sigma$
"Sorting By Reversals" Problem

A simplified restatement of the same problem...

- **Goal:** Given two permutations find the shortest series of reversals that transforms it into the identity permutations \(1, 2, 3, \ldots n\).
- **Input:** Permutations \(\Pi\)
- **Output:** A series of reversals \(\rho_1(s_1, e_1), \rho_2(s_2, e_2), \ldots \rho_t(s_t, e_t)\) transforming \(\Pi\) to the identity permutation such that \(t\) is minimal
- \(t = d(\Pi)\) - reversal distance of \(\Pi\)
Sorting By Reversals: Examples

\[ \pi_0 = 3, 4, 2, 1, 5, 6, 7, 10, 9, 8 \quad \rho(1, 2) \]
\[ \pi_1 = 4, 3, 2, 1, 5, 6, 7, 10, 9, 8 \quad \rho(8, 10) \]
\[ \pi_2 = 4, 3, 2, 1, 5, 6, 7, 8, 9, 10 \quad \rho(1, 4) \]
\[ \pi_3 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \]

\[ d(\pi) = 3 \]
Sorting By Reversals: Another example

\[ \Pi_0 = 2, 4, 3, 5, 8, 7, 6, 1 \quad \rho(2, 3) \]
\[ \Pi_1 = 2, 3, 4, 5, 8, 7, 6, 1 \quad \rho(5, 7) \]
\[ \Pi_2 = 2, 3, 4, 5, 6, 7, 8, 1 \quad \rho(1, 7) \]
\[ \Pi_3 = 8, 7, 6, 5, 4, 3, 2, 1 \quad \rho(1, 8) \]
\[ \Pi_4 = 1, 2, 3, 4, 5, 6, 7, 8 \]

\[ d(\Pi) = 4 \]

- But, is 4 the minimum number of reversals? Can it be done in 3?
Next Time

- Examples are not an algorithm
- Next time we'll discuss algorithms for performing reversal sorts