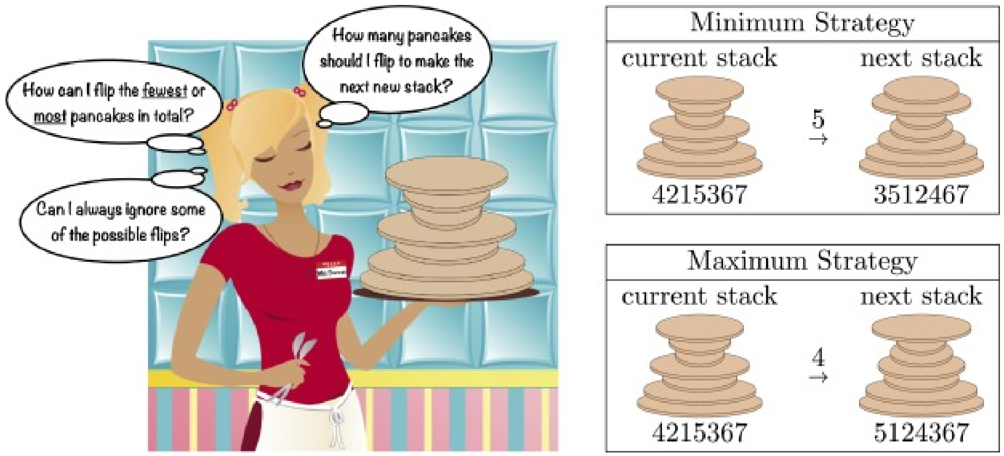
Genome Rearrangements - Continued



A Greedy Algorithm for Sorting by Reversals

- When sorting the permutation, $\Pi = 1, 2, 3, 6, 4, 5$, one notices that the first three elements are already in order.
- So it does not make sense to break them apart.
- The length of the already sorted prefix of Π is denoted as $prefix(\Pi) = 3$
- This inspires the following simple *greedy* algorithm

> while $prefix(\Pi) < len(\Pi)$:
>   perform a reversal $\rho(prefix(\Pi) + 1, k)$ such that $prefix(\Pi)$ increases by at least 1.

- Such a reversal must always exist
- Finding, k, is as simple as finding the index of the minimum value of the remaining unsorted part

Geedy Reversal Sort: Example

$$Step1 : \Pi_1 = 1, 2, 3, \underline{6, 4}, 5 \qquad \rho(4, 5)$$

$$Step2 : \Pi_2 = 1, 2, 3, 4, \underline{6, 5} \qquad \rho(5, 6)$$

$$Done : \Pi_3 = 1, 2, 3, 4, 5, 6$$

- The number of steps to sort any permuation of length n is at most $(n-1)$

Greedy Reversal Sort as code

```python
def GreedyReversalSort(pi):
    t = 0
    for i in xrange(len(pi)-1):
        j = pi.index(min(pi[i:]))
        if (j != i):
            pi = pi[:i] + [v for v in reversed(pi[i:j+1])] + pi[j+1:]
            print "rho(%2d,%2d) = %s" % (i+1,j+1,pi)
            t += 1
    return t

print GreedyReversalSort([3,4,2,1,5,6,7,10,9,8])
```

```
rho( 1, 4) = [1, 2, 4, 3, 5, 6, 7, 10, 9, 8]
rho( 3, 4) = [1, 2, 3, 4, 5, 6, 7, 10, 9, 8]
rho( 8,10) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
3
```

Analyzing GreedyReversalSort()

- GreedyReversalSort requires at most $n - 1$ steps
- For example, on $\Pi = 6, 1, 2, 3, 4, 5$, $t = 5$

$$\begin{aligned} \Pi_1 &: 6, 1, 2, 3, 4, 5 \\ \rho(1,2) &: 1, 6, 2, 3, 4, 5 \\ \rho(2,3) &: 1, 2, 6, 3, 4, 5 \\ \rho(3,4) &: 1, 2, 3, 6, 4, 5 \\ \rho(4,5) &: 1, 2, 3, 4, 6, 5 \\ \rho(5,6) &: 1, 2, 3, 4, 5, 6 \end{aligned}$$

- But there is a solution with far fewer flips

Greed Gone Wrong

- The same sequence sorted with two reversals

$$\Pi : 6, 1, 2, 3, 4, 5$$
$$\rho(1,6) : 5, 4, 3, 2, 1, 6$$
$$\rho(1,5) : 1, 2, 3, 4, 5, 6$$

- Note, this solution makes no progress (no elements of the permuation are placed in their correct order) after its first move
- Yet it beats a greedy approach handily.
- So SimpleReversalSort(π) is correct (as a sorting routine), but non-optimal
- For many problems there is no known optimal algorithm, in such cases *approximation algorithms* are often used.

Approximation Algorithms

- Today's algorithms find approximate solutions rather than optimal solutions
- The *approximation ratio* of an algorthim, $\mathcal{A}$ on input Π is:

$$r = \frac{\mathcal{A}(\Pi)}{OPT(\Pi)}$$

- where:
  - $\mathcal{A}(\Pi)$ is the number of steps using the given algorithm
  - $OPT(\Pi)$ is the number of steps required using, a possibly unknown, optimal algorithm

Performance Guarantees

- On an occasional input our approximation algorithm may give an optimal result, however we want to consider the value of r for the worst case input
- When our objective is to minimize something (like reversals in our case)

$$r = \max_{i=0}^{len(\Pi)!} \frac{\mathcal{A}(\Pi_i)}{OPT(\Pi_i)} \geq 1.0$$

- Or when our ojective is to maximize something (like money)

$$r = \max_{i=0}^{len(\Pi)!} \frac{\mathcal{A}(\Pi_i)}{OPT(\Pi_i)} \leq 1.0$$

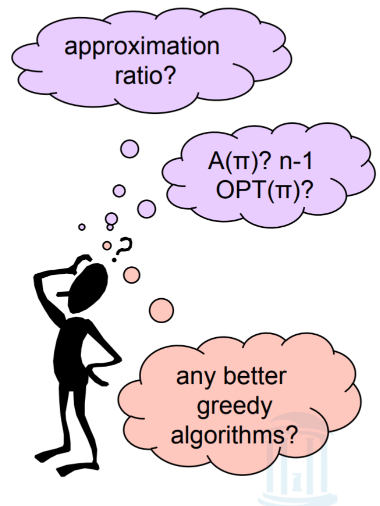
- Sounds cool in theory, but there are lots of open ends here
  - if we don't know $OPT(\Pi_i)$ how are we supposed to know how many steps it requires?
  - do we really need to test for all $len(\Pi)!$ possible inputs?

How do we get Approximation Ratios?

```
def GreedyReversalSort(pi):
    for i in xrange(len(pi)-1):
        j = pi.index(min(pi[i:]))
        if (j != i):
            pi = pi[:i]
                + [v for v in reversed(pi[i:j+1])]
                + pi[j+1:]
    return pi
```

```
        A(Π)
Step 0: 6 1 2 3 4 5
Step 1: 1 6 2 3 4 5
Step 2: 1 2 6 3 4 5
Step 3: 1 2 3 6 4 5
Step 4: 1 2 3 4 6 5
Step 5: 1 2 3 4 5 6
```

```
        OPT(Π)?
Step 0: 6 1 2 3 4 5
Step 1: 5 4 3 2 1 6
Step 2: 1 2 3 4 5 6
```



New Idea: Adjacencies

- Recall breakpoints from last lecture. Adjacencies are the opposite.
- Assume a permutation:

$$\Pi = \pi_1, \pi_2, \pi_3, \ldots \pi_{n-1}, \pi_n,$$

- A pair of neighboring elements π_i and π_{i+1} are *adjacent if:

$$\pi_{i+1} = \pi_i + 1$$

- For example:

$$\Pi = 1, 9, \underline{3, 4}, \underline{7, 8}, 2, \underline{6, 5}$$

- (3,4) and (7,8) and (6,5) are adjcencies.

Adjacencies and Breakpoints

- *Breakpoints* occure between neighboring non-adjacent elements

$$\Pi = 1, \color{red}{|}\, 9, \color{red}{|}\, \underline{3, 4}, \color{red}{|}\, \underline{7, 8}, \color{red}{|}\, 2, \color{red}{|}\, \underline{6, 5}$$

- There are 5 breakpints in our permuation between pairs (1,9), (9,3), (4,7), (8,2) and (2,5)

- We define $b(\Pi)$ as the number of breakpoints in permutation Π

First Previous Next Notebook

Extending Permutations

- One can place two elements, $\pi_0 = 0$ and $\pi_{n+1} = n + 1$ at the beginning and end of Π respectively

$$1, | 9, | \underline{3, 4,} | \underline{7, 8,} | 2, | \underline{6, 5}$$

$$\downarrow$$

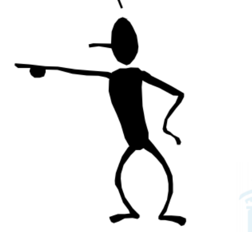
$$\Pi = 0\ 1, | 9, | \underline{3, 4,} | \underline{7, 8,} | 2, | \underline{6, 5,} | 10$$

- An addtional breakpoint was created after extending
- An extended permutation of length n can have at most $(n + 1)$ breakpoints
- $(n - 1)$ between the original elements plus 2 for the extending elements

How Reversals Effect Breakpoints

- Breakpoints are the *targets* for sorting by reversals.
- Once they are removed, the permutation is sorted.
- Each "useful" reversal eliminates at least 1, and at most 2 breakpoints.
- Consider the following application of GreedyReversalSort(Extend(Π))

$$\textit{required reversals} \geq \frac{b(\pi)}{2}$$



$$\Pi = \quad 2, 3, 1, 4, 6, 5$$

$$0 \mid 2, 3 \mid 1 \mid 4 \mid 6, 5 \mid 7 \qquad b(\Pi) = 5$$

$$0,\ 1 \mid 3, 2 \mid 4 \mid 6, 5 \mid 7 \qquad b(\Pi) = 4$$

$$0, 1, 2, 3, 4 \mid 6, 5 \mid 7 \qquad b(\Pi) = 2$$

$$0, 1, 2, 3, 4, 5, 6, 7 \qquad b(\Pi) = 0$$

Sorting By Reversals: A second Greedy Algorithm

<u>BreakpointReversalSort(π):</u>

1. while $b(\pi) > 0$:
2. Among all possible reversals, choose reversal ρ minimizing $b(\pi)$
3. $\Pi \leftarrow \Pi \cdot \rho(i, j)$
4. output Π
5. return

The "greedy" concept here is to reduce as many breakpoints as possible at each step.

Does it always terminate?

What if no reversal reduces the number of breakpoints?

0 1 2 | 5 6 7 | 3 4 | 8 9

New Concept: *Strips*

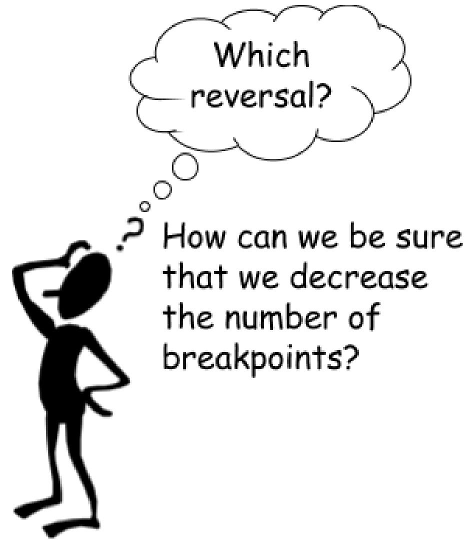
- ***Strip***: an interval between two consecutive breakpoints in a permutation
  - *Decreasing strip:* strip of elements in decreasing order (e.g. 6 5 and 3 2).
  - *Increasing strip:* strip of elements in increasing order (e.g. 7 8)
  - A single-element strip can be declared either increasing or decreasing.
  - We will choose to declare them as decreasing with exception of extension strips (with 0 and n+1)

$$\overrightarrow{0, 1}, \overleftarrow{9}, \overleftarrow{4, 3}, \overrightarrow{7, 8}, \overleftarrow{2}, \overrightarrow{5, 6}, \overrightarrow{10}$$

Reducing the Number of Breakpoints

- Consider $\Pi = 1, 4, 6, 5, 7, 8, 3, 2$

$$\overrightarrow{0, 1}, \mid \overleftarrow{4}, \mid \overleftarrow{6, 5}, \mid \overrightarrow{7, 8}, \mid \overleftarrow{3, 2}, \mid \overrightarrow{9} \qquad b(p) = 5$$



If permutation p contains at least one decreasing strip, then there exists a reversal r which decreases the number of breakpoints (i.e. $b(p \bullet r) < b(p)$).

Things to Consider

- Consider $\Pi = 1, 4, 6, 5, 7, 8, 3, 2$

$$\overrightarrow{0, 1}, \mid \underleftarrow{\overleftarrow{4}, \mid \overleftarrow{6, 5}, \mid \overrightarrow{7, 8}, \mid \overleftarrow{3, 2}}, \mid \overrightarrow{9} \qquad b(p) = 5$$

- Choose the decreassing strip with the smallest elment k in Π
  - It'll always be the rightmost element of that strip
- Find $k - 1$ in the permutation
  - it'll always be flanked by a breakpoint
- Reverse the segment between k and $k - 1$

Things to Consider

- Consider $\Pi = 1, 4, 6, 5, 7, 8, 3, 2$

$$\overrightarrow{0, 1, 2, 3}, \mid \overleftarrow{8, 7}, \mid \overrightarrow{5, 6}, \mid \overleftarrow{4}, \mid \overrightarrow{9} \qquad b(p) = 4$$

- Choose the decreassing strip with the smallest elment k in Π
  - It'll always be the rightmost element of that strip
- Find $k - 1$ in the permutation
  - it'll always be flanked by a breakpoint
- Reverse the segment between k and $k - 1$

Reversal Examples

- Consider $\Pi = 1, 4, 6, 5, 7, 8, 3, 2$

$$\overrightarrow{0, 1, 2, 3, 4,} \mid \underleftarrow{\overleftarrow{6, 5,}} \mid \overrightarrow{7, 8, 9} \qquad b(p) = 2$$

- Choose the decreasing strip with the smallest elment k in Π
  - It'll always be the rightmost element of that strip
- Find $k - 1$ in the permutation
  - it'll always be flanked by a breakpoint
- Reverse the segment between k and $k - 1$

Reversal Examples

- Consider $\Pi = 1, 4, 6, 5, 7, 8, 3, 2$

$$\overrightarrow{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} \qquad b(p) = 0$$

- Choose the decreasing strip with the smallest elment k in Π
  - It'll always be the rightmost element of that strip
- Find $k - 1$ in the permutation
  - it'll always be flanked by a breakpoint
- Reverse the segment between k and $k - 1$

Things to Consider

- Consider $\Pi = 1, 4, 6, 5, 7, 8, 3, 2$

$$\overrightarrow{0, 1}, \mid \overleftarrow{4}, \mid \overleftarrow{6, 5}, \mid \overrightarrow{7, 8}, \mid \overleftarrow{3, 2}, \mid \overrightarrow{9} \qquad b(p) = 5$$

$$\overrightarrow{0, 1, 2, 3}, \mid \overleftarrow{8, 7}, \mid \overrightarrow{5, 6}, \mid \overleftarrow{4}, \mid \overrightarrow{9} \qquad b(p) = 4$$

$$\overrightarrow{0, 1, 2, 3, 4}, \mid \overleftarrow{6, 5}, \mid \overrightarrow{7, 8, 9} \qquad b(p) = 2$$

$$\overrightarrow{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} \qquad b(p) = 0$$

$$d(\Pi) = 3$$

Potential Gotcha

$$\overrightarrow{0, 1, 2,} \mid \overrightarrow{5, 6, 7,} \mid \overrightarrow{3, 4,} \mid \overrightarrow{8, 9} \qquad b(p) = 3$$

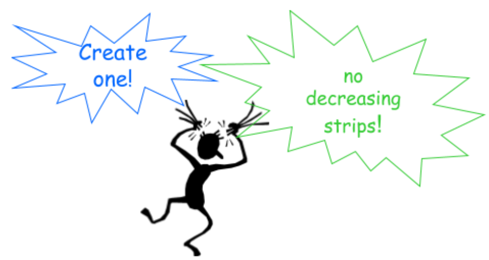
- If there is no decreasing strip, there may be *no strip-reversal* ρ *that reduces the number of breakpoints* (i.e. $b(\Pi \cdot \rho(i, j)) \geq b(\Pi)$ for any reversal ρ).
- However, reversing an increasing strip creates a decreasing strip, and the number of breakpoints remains unchanged.
- Then the number of breakpoints will be reduced in the following steps.

Potential Gotcha

$$\overrightarrow{0, 1, 2}, \mid \underleftarrow{\overrightarrow{5, 6, 7}}, \mid \overrightarrow{3, 4}, \mid \overrightarrow{8, 9} \qquad b(p) = 3$$

$$\overrightarrow{0, 1, 2}, \mid \overleftarrow{7, 6, 5}, \mid \overrightarrow{3, 4}, \mid \overrightarrow{8, 9} \qquad b(p) = 3$$

- If there is no decreasing strip, there may be *no strip-reversal* ρ *that reduces the number of breakpoints* (i.e. $b(\Pi \cdot \rho(i,j)) \geq b(\Pi)$ for any reversal ρ).
- However, reversing an increasing strip creates a decreasing strip, and the number of breakpoints remains unchanged.
- Then the number of breakpoints will be reduced in the following steps.



Next Time

- Look at the Code
- How about performance?