**DNA Sequence Reconstruction**

- DNA can only be sequenced in relatively small pieces, up to about 1,000 nucleotides.
- By chemistry a much longer DNA sequence can be broken up into overlapping sequences called clones. Clones are 10's of thousands of nucleotides long.

```
DNA

clones

PQ-trees
```

**Tagging the Clones**

- By chemistry the clones can be tagged by identifying a region of the DNA uniquely.

```
DNA

 clones
```

- Each clone is then tagged correspondingly.

```
DNA

 clones
```

**Problem to Solve**

- Given a set of tagged clones, find a consistent ordering of the tags that determines a possible ordering of the DNA molecule.

<table>
<thead>
<tr>
<th>clone</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{E, G}</td>
</tr>
<tr>
<td>2</td>
<td>{F, G, H}</td>
</tr>
<tr>
<td>3</td>
<td>{A, I}</td>
</tr>
<tr>
<td>4</td>
<td>{C, D}</td>
</tr>
<tr>
<td>5</td>
<td>{E, G}</td>
</tr>
<tr>
<td>6</td>
<td>{A, H, I}</td>
</tr>
<tr>
<td>7</td>
<td>{B, D}</td>
</tr>
<tr>
<td>8</td>
<td>{F, H}</td>
</tr>
<tr>
<td>9</td>
<td>{A, B, D, I}</td>
</tr>
<tr>
<td>10</td>
<td>{C, D}</td>
</tr>
</tbody>
</table>

```
input  output
1      E      5
2      G      F
3      F      I
4      H      A
5      A      B
6      I      D
7      B      C
8      D
9      C
10     D
```

```
PQ-trees
```
Contiguous Ordering Solutions

Contiguous ordering problem
U = {A, B, C, D, E, F, G, H, I}

Solution
E G F H A B D C

Alternate Solutions
interchange I and A
reversal
E G F H A B D C
C D B I A H F G E
C D B A I H F G E

Linear Time Algorithm

- Booth and Lueker, 1976, designed an algorithm that runs in time $O(n+m+s)$.
  - $n$ is the size of the universe, $m$ is the number of sets, and $s$ is the sum of the sizes of the sets.
- It requires a novel data structure called the PQ tree that represents a set of orderings.
- PQ trees can also be used to test whether an undirected graph is planar.

PQ Trees

- PQ trees are built from three types of nodes
  - P node
  - Q node
  - leaf

- Children can be reordered.
- Children can be reversed.
- Each leaf has a unique label.

Example PQ-Tree

The frontier of T defines the ordering $F(T) = FCABDE$, just read the leaves left to right.

$T'$ is equivalent to T if T can be transformed into $T'$ by reordering the children of P nodes and reversing the children of Q nodes.
Equivalent PQ Trees

\[ \text{FCABDE} \quad \text{FEBDAC} \]

Orderings Defined by a PQ Tree

- Given a PQ tree \( T \), the orderings defined by \( T \) is
  \[ \text{PQ}(T) = \{ F(T') : T' \text{ is equivalent to } T \} \]

There are 6 x 2 x 2 = 24 distinct orderings in \( \text{PQ}(T) \).

Generally, if a PQ tree \( T \) has \( q \) Q node and \( p \) P nodes with number of children \( c_1, c_2, \ldots, c_p \), then the number of orderings in \( \text{PQ}(T) \) is
\[ 2^q c_1! c_2! \ldots c_p! \]

\[ n! = 1 \times 2 \times \ldots \times n \]

PQ Tree Solution for the Contiguous Ordering Problem

- Input: A universe \( U \) and a set \( S = \{ S_1, S_2, \ldots, S_m \} \) of subsets of \( U \).
- Output: A PQ tree \( T \) with leaves \( U \) with the property that \( \text{PQ}(T) \) is the set of all orderings of \( U \) where each set in \( S \) is contiguous in the ordering.

Example Solution

\[ U = \{A,B,C,D,E,F\} \]
\[ S = \{(A,C,E), (A,C,F), (B,D,E)\} \]

There are 8 orderings that are possible in keeping each of these sets contiguous.
PQ Tree Restriction

- Let $U = \{A_1, A_2, \ldots, A_n\}$, $S = \{A_1, A_2, \ldots, A_k\}$, and $T$ a PQ tree.
- We will define a function $\text{Restrict}$ with the following properties:
  - $\text{Restrict}(T, S)$ is a PQ tree.
  - $\text{PQ}(\text{Restrict}(T, S)) = \text{PQ}(T) \cap \text{PQ}(T')$ where

High Level PQ tree Algorithm

- Input is $U = \{A_1, A_2, \ldots, A_n\}$, and subsets $S_1, S_2, \ldots, S_m$ of $U$.
- Initialization:
  - $T = \text{P node with children } A_1, A_2, \ldots, A_n$
- Calculate $m$ restrictions:
  - for $j = 1$ to $m$ do
    - $T := \text{Restrict}(T, S_j)$
- At the end of iteration $k$:
  - $\text{PQ}(T) =$ the set of ordering of $U$ where each set $S_1, S_2, \ldots, S_k$ are contiguous.

Marking Nodes

- Given a set $S$ and PQ tree $T$ we can mark nodes either full or partial.
  - A leaf is full if it is a member of $S$.
  - A node is full if all its children are full.
  - A node is partial if either it has both full and non-full children or it has a partial child.
  - A node is doubly partial if it has two partial children.

Marks of Nodes

Mark the leaves in $S$ full. Bottom up mark the nodes full or partial. The members of $S$ will become contiguous.
Structure of the Marked PQ Tree

Restrict(T,S)
- Mark the full and partial nodes from the bottom up.
  - In the process the marked leaves become contiguous.
- Locate the key node.
  - Deepest node with the property that all the full leaves are descendents of the node.
- Restrict the key node.
  - In the process of restricting the key node we will have to recursively direct partial nodes.
  - Directing a node returns a sequence of nodes.

Restricting a P Node with Partial Children

Restricting a P node with no Partial Children
Restricting a Q node

Directing a P Node

Directing a Q Node

Example (1)

\[ U = \{A,B,C,D,E,F,G,H,I,J\} \]
\[ S_1 = \{A,C,E,G,I\} \]

mark
Example (2)

U = \{A,B,C,D,E,F,G,H,I,J\}
S_1 = \{A,C,E,G,I\}

restrict P node

special case because no partial child.

Example (3)

U = \{A,B,C,D,E,F,G,H,I,J\}
S_2 = \{C,D,F,G,I,J\}

mark

Example (4)

U = \{A,B,C,D,E,F,G,H,I,J\}
S_2 = \{C,D,F,G,I,J\}

restrict P node

Example (5)

U = \{A,B,C,D,E,F,G,H,I,J\}
S_2 = \{C,D,F,G,I,J\}

direct P node
Example (6)

\[ U = \{A, B, C, D, E, F, G, H, I, J\} \]
\[ S_2 = \{C, D, F, G, I, J\} \]

Example (7)

\[ U = \{A, B, C, D, E, F, G, H, I, J\} \]
\[ S_3 = \{A, B, E, G\} \]

Example (8)

\[ U = \{A, B, C, D, E, F, G, H, I, J\} \]
\[ S_3 = \{A, B, E, G\} \]

Example (9)

\[ U = \{A, B, C, D, E, F, G, H, I, J\} \]
\[ S_3 = \{A, B, E, G\} \]
Example (10)

\[ U = \{A,B,C,D,E,F,G,H,I,J\} \]
\[ S_3 = \{A,B,E,G\} \]

Example (11)

\[ U = \{A,B,C,D,E,F,G,H,I,J\} \]
\[ S_3 = \{A,B,E,G\} \]

Example (12)

\[ U = \{A,B,C,D,E,F,G,H,I,J\} \]
\[ S_1 = \{A,C,E,G,I\} \]
\[ S_2 = \{C,D,F,G,I,J\} \]
\[ S_3 = \{A,B,E,G\} \]

Exercise

- Restrict with to make \{A,B,E,G\} contiguous
Linear Number of Nodes Processed

- Let \( n \) be the size of the universe, \( m \) the number of sets, and \( s \) the sum of the sizes of the sets.
  - Number of full nodes processed < 2s.
  - Number of key nodes processed = \( m \).
  - Number of partial nodes with partial children processed below the key node < \( m + n \).
  - Number of partial nodes with no partial children < \( 2m \).
  - Number of partial nodes processed above the key node < \( m + n \).

Partials with Partial Children Below the Key Node

- Amortized complexity argument.
- Consider the quantities:
  - \( q \) = number of Q nodes,
  - \( cp \) = number of children of P nodes.
  - We examine the quantity \( x = q + cp \)
  - \( x \) is initially \( n \) and never negative.
  - Each restrict of a key node increases \( x \) by at most 1.
  - Each direct of a partial node with a partial child decreases \( x \) by at least 1.
  - Since there are \( m \) restricts of a key node then there are most \( n + m \) directs of partials with partial children.

Number of Processed Nodes Amortized

- Key Nodes \( m \)
- Partial nodes with exactly one partial child \( \leq m + n \)
- Partial nodes with partial children \( \leq m + n \)
- Full \( \leq 2s \)
- Partial nodes with no partial children \( \leq 2m \)

Restricting a P Node with Partial Children

- Restrict a P node
- Change in \( q + cp \) is at most +1.
Restricting a P node with no Partial Children

Restricting a Q node

Directing a P Node

Directing a Q Node
PQ Tree Notes

- In algorithmic design only a linear number of nodes are ever processed.
- Designing the data structures to make the linear time processing a reality is very tricky.
- PQ trees solve the idealized DNA ordering problem.
- In reality, because of errors, the DNA ordering problem is NP-hard and other techniques are used.

Example of Data Structure Trick

- Linking the children of a Q node

![Diagram of PQ trees](image)