Reconstructing the suffix array

F. Franek & B. Smyth

Algorithms Research Group
Dept. of Computing & Software
McMaster University
Hamilton, Ontario

LSD’05, February 2005
Recently, we came across an interesting problem: knowing the order of suffixes of a binary string, can one “easily” tell the order of suffixes of its complement?

“Obvious” answer --- just reverse it --- is wrong!

\[
\begin{align*}
\text{aaba} & \quad < \quad \text{aaba} < \quad \text{aba} < \quad \text{ba} \\
\text{aaba} = \text{bbab} & \quad < \quad \text{bab} < \quad \text{bbab}
\end{align*}
\]

The reason: if \( p_1 < p_2 \) are suffixes, either

- \( p_1 \) is a prefix of \( p_2 \), and then \( p_1 < p_2 \)
- \( p_1 \) is not a prefix of \( p_2 \), and then \( p_1 > p_2 \)
The problem can be phrased in a more general way:

knowing the order of suffixes of a string, if we reverse the order of the alphabet, can one “easily” tell the new order of suffixes?

What do we mean by “easily”? Usually a linear-time algorithm.
A simple answer:

- knowing the order of suffixes, in linear time compute the lcp info \((Kasai \ et \ al, \ 2001)\) and build the suffix array
- having the suffix array, build the suffix tree
- invert the order of the links in every node of the suffix tree
- traverse the suffix tree in depth-first (inorder) fashion and “read out” the new order of suffixes.
The problem with this solution: a suffix tree requires between 5N to 10N words of memory (N the length of the string), on top of this you need some working memory.

So, what do we really mean by “easily”? A linear-time and a memory-efficient algorithm.

We designed a linear-time iterative (non-recursive) algorithm that requires a working memory of 2N words and that re-sorts suffixes of a string after reversing the order of the alphabet.
The algorithm is based on the observation that the order of suffixes will be reversed as well, with the exception of prefixes, for which the order will be preserved. So, we need to be able to determine for each pair of suffixes $p_1 < p_2$ whether $p_1$ is a prefix of $p_2$ or not.

For that purpose we will compute in essence a reverse border array:

$$rba[i]=j \quad \text{iff} \quad x[i..N] \text{ has } x[j..N] \text{ as a maximal border.}$$

A simple modification of the failure function algorithm (Aho, Hopcroft, Ullman, 1974) can compute $rba[ ]$ in linear time.
Algorithm 1

- let $s_1..s_N$ be the suffixes sorted according to the original order of the alphabet
- let $t_1..t_{k-1}$ be $s_1..s_{k-1}$ sorted according to the reversed order of the alphabet, let $s_k=x[p..N]$
- if $rba[p]=null$ then move $s_k$ to the front of $t_1..t_{k-1}$
- if $rba[p] \neq null$ then move $s_k$ right behind $tr$ where $tr=x[rba[p]..N]$

Let us run through an example for illustration:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| a | b | c | a | a | a | b | c | d | a | b | c | c | a | b | d | a | b | e | a |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 18 | 3 | 0 | 8 | 4 | 12 | 15 | 1 | 9 | 5 | 13 | 16 | 2 | 11 | 10 | 6 | 7 | 14 | 17 |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 8 | 9 | 11 | 0 | 12 | 13 | 7 | 14 | 4 | 5 | 6 | 10 | 15 | 16 | 17 | 1 | 2 | Ø | 3 |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 3 | 15 | 16 | 18 | 8 | 9 | 10 | 6 | 0 | 1 | 11 | 2 | 4 | 5 | 7 | 12 | 13 | 14 | Ø |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 18 | Ø | Ø | 18 | 18 | Ø | Ø | Ø | 18 | Ø | Ø | Ø | 18 | Ø | Ø | 18 | Ø | Ø | Ø |

Slide 10
Slide 11
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 3 | 9 | 11 | 8 | 12 | 13 | 7 | 14 | 4 | 5 | 6 | 10 | 15 | 16 | 17 | 1 | 2 | ∅ | ∅ | 0 |
| 18 | 15 | 16 | 0 | 8 | 9 | 10 | 6 | 0 | 1 | 11 | 2 | 4 | 5 | 7 | 12 | 13 | 14 | ∅ |
| ∅ | ∅ | 18 | 18 | ∅ | ∅ | ∅ | 18 | ∅ | ∅ | ∅ | 18 | ∅ | ∅ | 18 | ∅ | ∅ | ∅ | ∅ |

Slide 12
Slide 13
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>9</td>
<td>11</td>
<td>1</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>16</td>
<td>17</td>
<td>12</td>
<td>2</td>
<td>Ø</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>3</td>
<td>16</td>
<td>0</td>
<td>12</td>
<td>9</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>15</td>
<td>5</td>
<td>7</td>
<td>18</td>
<td>13</td>
<td>14</td>
<td>Ø</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>Ø</td>
<td>Ø</td>
<td>18</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>18</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>18</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>18</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>18</th>
<th>15</th>
<th>12</th>
<th>4</th>
<th>8</th>
<th>0</th>
<th>3</th>
<th>1</th>
<th>9</th>
<th>5</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>2</td>
<td>11</td>
<td>10</td>
<td>6</td>
<td>7</td>
<td>14</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|
| 0 | 3 | 18| 11| 9 | 8 | 13| 7 | 14| 0 | 5 | 6  | 10 | 4  | 16| 17 | 12| 2  | Ø  | 15 |
| 1 | 8 | Ø | 16| 0 | 12| 9 | 10| 6 | 4 | 3  | 11 | 2  | 15 | 5  | 7 | 18 | 13| 14 | 1  |
| 2 | 18| Ø | Ø | 18| Ø | Ø | Ø | 18| Ø | Ø | Ø  | 18 | Ø  | Ø  | Ø  | 18| Ø  | Ø  | Ø  | Ø  |

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>18</th>
<th>15</th>
<th>12</th>
<th>4</th>
<th>8</th>
<th>0</th>
<th>3</th>
<th>1</th>
<th>9</th>
<th>5</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>2</td>
<td>11</td>
<td>10</td>
<td>6</td>
<td>7</td>
<td>14</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Slide 16
What about an arbitrary permutation of the order of the alphabet?

There we run into several problems. Again, for a suffix tree all we have to worry about is the permutation itself, i.e. how complicated it is to sort each “family” of suffixes (resorting the families of links may not be linear). More about the complexity of permutations later.

If we start with a suffix array of a string, rather than just an ordered sequence of suffixes as we did for the inversion of the alphabet, we can identify and sort “families” as in suffix tree. This “extra” work can be done in linear-time with $\leq 2N$ words of working memory.

The iterative (non-recursive) algorithm relies on four steps repeated until the end of processing. These four steps are: identify-and-extract family, sort the family, flatten the family, and verticalize the family. The following example will illustrate these procedures:
We are using again the same string as in Slide 6. The permutation of the alphabet for this examples is defined by

\[
\begin{align*}
  p[a] &= a \\
  p[b] &= c \\
  p[c] &= d \\
  p[d] &= b \\
  p[e] &= e
\end{align*}
\]

\[
\begin{array}{cccccccccccccccc}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\
  a & b & c & a & a & b & c & d & a & b & c & c & a & b & d & a & b & e & a
\end{array}
\]

\[
\begin{array}{cccccccccccccccc}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\
  18 & 3 & 0 & 8 & 4 & 12 & 15 & 1 & 9 & 5 & 13 & 16 & 2 & 11 & 10 & 6 & 7 & 14 & 17 \\
  1 & 1 & 3 & 3 & 2 & 2 & 0 & 2 & 2 & 1 & 1 & 0 & 2 & 1 & 1 & 0 & 3 & 0
\end{array}
\]
<table>
<thead>
<tr>
<th>lcp</th>
<th>next</th>
<th>tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

18 (1)
stack
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>1</td>
<td>2</td>
<td>Ø</td>
<td>3</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

18 → 3 → 0 → 8 → 4 → 12 → 15 → 1 → 9

5 → 13 → 16 → 2 → 11 → 10 → 6 → 7 → 14 → 17

0 (3)
18 (1)

stack
identify and extract family

3-family identify and extract family

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
3 2 2 1 2 1 0 3 3 2 1 1 2 1 0 0 0 Ø 1
8 9 11 0 Ø 13 7 14 4 5 6 10 15 16 17 1 2 Ø 3
Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø

18 → 3 → 0 → 8 → 4 → 12 → 15 → 1 → 9

5 → 13 → 16 → 2 → 11 → 10 → 6 → 7 → 14 → 17

0 (3)
18 (1)

3-family

3 3 2
0 8 4

Slide 25
sort the family

<table>
<thead>
<tr>
<th>Stack</th>
<th>3-family</th>
<th>LCP &quot;sort&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (3)</td>
<td>3 3 2</td>
<td>1. last is preserved</td>
</tr>
<tr>
<td>18 (1)</td>
<td>0 8 4</td>
<td>2. every &gt;= k preserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. everything else is k</td>
</tr>
</tbody>
</table>

1. 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
2. 3 2 2 1 3 1 0 3 2 2 1 1 2 1 0 0 0 Ø 1
3. 4 9 11 0 8 13 7 14 Ø 5 6 10 15 16 17 1 2 Ø 3
4. Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø

18 -- 3 -- 0 -- 4 -- 8 -- 12 -- 15 -- 1 -- 9
5 -- 13 -- 16 -- 2 -- 11 -- 10 -- 6 -- 7 -- 14 -- 17
flatten the family

nothing to flatten, it is already flat

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>11</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>Ø</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>1</td>
<td>2</td>
<td>Ø</td>
<td>3</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

18 → 3 → 0 → 4 → 8 → 12 → 15 → 1 → 9

5 → 13 → 16 → 2 → 11 → 10 → 6 → 7 → 14 → 17
verticalize the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>Ø</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>1</td>
<td>2</td>
<td>Ø</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>8</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

Slide 28
identify and extract family

2-family

stack

%28
sort the family

2-family

<table>
<thead>
<tr>
<th>0 (3)</th>
<th>18 (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>stack</td>
</tr>
</tbody>
</table>

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
3 2 2 1 3 1 0 3 2 2 1 1 2 1 0 0 0 Ø 1
15 9 11 12 8 13 7 14 Ø 5 6 10 0 16 17 Ø 2 Ø 3
4 Ø Ø Ø 8 Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø 0 8 4

tailnext  tailend

18 → 3 → 12 → 0 → 15 → 1 → 9 → 5 → 13 → 16 → 2

11 → 10 → 6 → 7 → 14 → 17
### flatten the family

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|
| 3 | 2 | 2 | 1 | 3 | 1 | 0 | 3 | 2 | 2 | 1 | 1   | 2   | 1   | 0   | 0   | 0   | Ø   | 1   |
| 4 | 9 | 11 | 12 | 8 | 13 | 7 | 14 | 15 | 5 | 6 | 10   | 16  | 17  | Ø   | 2   | Ø   | 3   |
| Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø   | Ø   | Ø   | Ø   | Ø   | Ø   | Ø   | Ø   |

![Diagram of tree flattening](image-url)
verticalize the family

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 3 | 2 | 2 | 1 | 3 | 1 | 0 | 3 | 2 | 2 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | Ø | 1 |
| 4 | 9 | 11 | 12 | 8 | 13 | 7 | 14 | 15 | 5 | 6 | 10 | 1 | 16 | 17 | Ø | 2 | Ø | 3 |
| 15 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 0 | Ø | Ø | Ø | Ø | Ø |

tailend ![Diagram showing the family verticalization](image)

tailnext
identify and extract family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>Ø</td>
<td>16</td>
<td>17</td>
<td>Ø</td>
<td>2</td>
<td>Ø</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

15 | 8 | 4 | 0 | 18 | 3 | 12 | 1 | 9 | 5 | 13 | 16 | 2 | 11 |

18 (1)

stack

1-family

1 | 1 | 3

18 3 12

Slide 33
sort the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>Ø</td>
<td>16</td>
<td>17</td>
<td>Ø</td>
<td>2</td>
<td>Ø</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

sort the family

18 (1) stack

1-family

1 1 3

18 3 12

1 1 3

18 3 12

Slide 34
### flatten the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>16</td>
<td>17</td>
<td>Ø 2</td>
<td>Ø 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td></td>
</tr>
</tbody>
</table>

---

18 → 3 → 12 → 0 → 4 → 8 → 15 → 1 → 9

5 → 13 → 16 → 2 → 11 → 10 → 6 → 7 → 14 → 17

---

Slide 35
verticalize the family

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>16</td>
<td>17</td>
<td>Ø</td>
<td>2</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>
```

```
18 → 1 → 9 → 5 → 13 → 16 → 2 → 11 → 10 → 6
```

```
15

8

4

0

12

3

7 → 14 → 17
```

Slide 36
Identify and extract family.
sort the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>Ø</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>16</td>
<td>17</td>
<td>Ø</td>
<td>2</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>

tailend

tailnext

1 (2)
stack

2-family

<table>
<thead>
<tr>
<th>2</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

→

<table>
<thead>
<tr>
<th>2</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>

18 → 1 → 5 → 9 → 13 → 16 → 2 → 11 → 10 → 6

7 → 14 → 17
nothing to flatten, it is already flat
verticalize the family

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>Ø</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Ø</td>
<td></td>
<td>5</td>
<td></td>
<td>9</td>
<td>Ø</td>
<td>9</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
</tbody>
</table>

Diagram labels:
- tailnext
- tailend
identify and extract family

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>Ø</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>16</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Ø</td>
<td>5</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>9</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Stack**: 1(2)

**1-family**: 2 1 0

1 13 16

15 8 4 0 12 9 3 5

18 1 13 16 2 11 10 6 7 14 17

Slide 41
### Sort the Family

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>16</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>Ø</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>Ø</td>
<td>5</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>9</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>

**Stack**

1(2)

**1-family**

2 1 0

1 13 16

13 1 16

---

Slide 42
flatten the family

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
3 2 2 1 3 2 0 3 2 1 1 1 2 1 0 0 0 Ø 1
4 5 11 12 8 9 7 14 15 16 6 10 0 1 17 Ø Ø Ø 13
Ø Ø Ø 15 Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø 3
```

tailend
tailnext

18 → 13 → 1 → 5 → 9 → 16 → 2 → 11 → 10 → 6
7 → 14 → 17
verticalize the family

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ø</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>1</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

15  
↓  
8  
↓  
4  
↓  
0  
↓  
12 
↓  
3  
↓  
1  
↓  
18 
→  
13 
→  
2  
→  
11 
→  
10 
→  
6  
→  
7  
→  
14 
→  
17

Slide 44
identify and extract family

2-family

stack

Slide 45
sort the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>6</td>
<td>Ø</td>
<td>0</td>
<td>2</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>0</td>
<td>1</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

15 — 8 — 4 — 16 — 12 — 3 — 18

2(2)

Stack

2-family

2

1

11

10

6 — 7 — 14 — 17

Slide 46
nothing to flatten, it is already flat

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>6</td>
<td>Ø</td>
<td>0</td>
<td>2</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>Ø</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>

```
15
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
  8
|   |   |
  4
|   |   |
  0
|   |   |
 12
|   |   |
  5
|   |   |
  3
|   |   |
 18
```

```
11
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
 13
|   |   |
  2
|   |   |
 10
|   |   |
  6
|   |   |
  7
|   |   |
 14
|   |   |
 17
```
verticalize the family

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 3 | 2 | 2 | 1 | 3 | 2 | 0 | 3 | 2 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | Ø | 1 |
| 4 | 5 | 10 | 12 | 8 | 9 | 7 | 14 | 15 | 16 | 6 | Ø | 0 | 2 | 17 | Ø | Ø | Ø | 13 |
| Ø | 16 | 11 | 15 | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 1 | Ø | Ø | Ø | Ø | 3 |

tail | tailnext | tailend | tail | tailend | tailnext | tailnext
identify and extract family
sort the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>Ø</td>
<td>Ø</td>
<td>0</td>
<td>2</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>11</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>1</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>

1-family

stack

2(2)

1-family

2 1 0

2 10 6

2 6 10

Slide 50
flatten the family

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|
| 3 | 2 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 0   | 1   | 2   | 1   | 0   | 0   | 0   | Ø   | 1   |
| 4 | 5 | 11| 12| 8 | 9 | 10| 14| 15| 16| Ø   | 6   | 0   | 2   | 17  | Ø   | Ø   | Ø   | 13  |
| Ø | 16| Ø | 15| Ø | Ø | Ø | Ø | Ø | Ø | Ø   | Ø   | 1   | Ø   | Ø   | Ø   | Ø   | Ø   | 3   |

tailend  tailend  tailnext  tailnext

15
8
4
0
12
3
18
13
2
11
6
10
7
14
17
verticalize the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>Ø</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>17</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>11</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>10</td>
<td>Ø</td>
<td>1</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>

tailend  tailnext tailend  tailend  tailnext  tailnext
identify and extract family

Slide 53
#### Slide 54

**Sort the family**

|    0  |    1  |    2  |    3  |    4  |    5  |    6  |    7  |    8  |    9  |   10  |   11  |   12  |   13  |   14  |   15  |   16  |   17  |   18  |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|  3   |  2   |  2   |  1   |  3   |  2   |  1   |  3   |  2   |  1   |   0   |   1   |   2   |   1   |   0   |   0   |   0   |   Ø   |   1   |
|  4   |  5   |  7   |  12  |  8   |  9   |  10  |  14  |  15  |  16  |  Ø    |   6   |  0    |   2   |  Ø    |  Ø    |  Ø    |  Ø    |  Ø    |  13   |
|  Ø   |  16  |  11  |  15  |  Ø   |  Ø   |  Ø   |  Ø   |  Ø   |  Ø   |  10   |  Ø    |  1    |  Ø    |  Ø    |  Ø    |  Ø    |  Ø    |  3    |

---

**Diagram:**

- **Stack:** 7(3)
- **3-family:**
  - 3
  - 0
  - 7
  - 14
- **Result:**
  - 3
  - 0
  - 7
  - 14
nothing to flatten, it is already flat

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>Ø</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>11</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>10</td>
<td>Ø</td>
<td>1</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**flatten the family**
verticalize the family

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Ø</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>17</td>
<td>15</td>
<td>16</td>
<td>Ø</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>13</td>
</tr>
<tr>
<td>Ø</td>
<td>16</td>
<td>11</td>
<td>15</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>14</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>10</td>
<td>Ø</td>
<td>1</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>3</td>
</tr>
</tbody>
</table>

tailend tailnext tailend tailnext tailend tailend tailend tailend tailnext tailend tailnext tailend tailnext tailend tailnext tailend tailend tailnext tailend tailnext
identify and extract family

Slide 57
sort the family

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 3 | 2 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | Ø | 1 |
| 4 | 5 | 7 | 12 | 8 | 9 | 10 | 17 | 15 | 16 | Ø | 6 | 0 | 2 | Ø | Ø | Ø | Ø | 13 |
| Ø | 16 | 11 | 15 | Ø | Ø | Ø | 14 | Ø | Ø | Ø | 10 | Ø | 1 | Ø | Ø | Ø | Ø | 3 |

stack 0-family

1 1 2 3 Ø
18 13 2 7 17

1 3 1 2 Ø
18 7 13 2 17

Slide 58
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| 3 | 2 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | Ø | 1 |
| 4 | 5 | 11 | 12 | 8 | 9 | 10 | 14 | 15 | 16 | 17 | 6 | 0 | 1 | 13 | 7 | 2 | Ø | 3 |
| Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10 | Ø | Ø | Ø | Ø | Ø | Ø |

- flatten the family

18 → 3 → 12 → 0 → 4 → 8 → 15 → 7 → 14 → 13

1 → 5 → 9 → 16 → 2 → 11 → 6 → 10 → 17
# Slide 60

Transform to suffix array

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3 | 2 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | Ø | 1 |
| 4 | 5 | 11 | 12 | 8 | 9 | 10 | 14 | 15 | 16 | 17 | 6 | 0 | 1 | 13 | 7 | 2 | Ø | 3 |
| Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø | 10 | Ø | Ø | Ø | Ø | Ø | Ø | Ø |

18 → 3 → 12 → 0 → 4 → 8 → 15 → 7 → 14 → 13
1 → 5 → 9 → 16 → 2 → 11 → 6 → 10 → 17

stop
The algorithm just presented shows yet again, that for most of applications suffix array is as efficient as suffix tree, yet requiring significantly less memory.

So, when the resorting process is linear?

1. Of course, when the alphabet is fixed, then the resorting is linear.

2. When the permutation is not too complex, then the resorting will also be linear.

Let us introduce the suborder complexity $\beta$ of a permutation $p$ of length $N$: $\beta(p) = \min \beta$ so that for any $2 \leq k \leq N$, it takes at most $\beta k$ steps to order any subset of $N$ of size $k$.

Note: $\beta(p) \leq \log N$ as any subset of $N$ of size $k$ can be sorted in $\leq k \log k$ steps and $k \log k \leq k \log N$
For any permutation with suborder complexity $\beta$, the suffix array of a string can be re-ordered in a $O(\beta N)$ time, where $N$ is the length of the input string.

- For instance, the inversion has suborder complexity of 1.
- Any rotation has suborder complexity of 1.
- Any permutation with $\beta$ transpositions has suborder complexity of $\beta$.
- Let $p$ be a “mild” permutation, i.e. $|p(i) - i| \leq \beta$. Then $p$ has suborder complexity of $2\beta$.
- Let $p_1$ on $N_1$ have suborder complexity $\beta_1$ and let $p_2$ on $N_2$ have suborder complexity $\beta_2$, then $p_1 \oplus p_2$ will have suborder complexity $\max(\beta_1, \beta_2)$. 
So, there are quite a few of permutations that allow us to re-sort the suffix array or the suffix tree of a string in linear time. 

*May be, it is of independent interest to study the suborder complexity of permutations.*

It will be also interesting to see, if it is more efficient to simply re-sort the suffixes, or if re-sorting it our way is more efficient. If our approach turns out to be more efficient, it may be conceivable to compute efficiently a suffix array according to some other order of the alphabet more conducive to the task and then re-sort it according to the natural order.

This is one of the possibilities we will be pursuing in our quest for a non-recursive linear-time and memory efficient algorithm to sort suffixes.
http://www.cas.mcmaster.ca/~franek