Lecture Problem I, part 3

Given Wed, June 3, 2020
Thurs, June 4, 2020
Fri, June 5, 2020

(PDF last updated Mon, July 6, 2020)

- corrections on pg 7 for the five permutations in S_5 which has bad pattern but whose soliton shape = RS shape
- Use "soliton decomposition" instead of "soliton content"

Ref

- Sagan Ch 3
- Fukuda " Box-ball systems and Robinson-Schensted-Knuth Correspondence" 2001-2004
- Wikipidea "permutation pattern"

Rose Checked w/ computer up to n=8, for any permutation $\pi \in S_n$ you can find a sequence of decompositions of π so that the i-decomposition contains the $(i+1)$-decomposition, for all $i = 1, \ldots, k$.

$\pi = \pi_1 \ldots \pi_n \qquad D_1^L$

$\pi = \pi_1 \pi_i \underset{i}{|} \ldots \pi_n \qquad D_2^L$

$\pi = \pi_1 \ldots \pi_i \underset{i}{|} \pi_{i+1} \ldots \underset{j}{|} \ldots \pi_n \qquad D_3^L$ where $j \neq i$

$\pi = \pi_1 | \ldots \pi_i \underset{i}{|} \pi_{i+1} \ldots \underset{j}{|} \ldots \pi_n \qquad D_4^L$

where k, i, j.

If you write down an algorithm, write it (also) for increasing subsequences

Try to find an algorithm to prove this.

$\pi = \pi_1 \underset{b_1}{|} \ldots | \ldots \underset{b_{k-1}}{|} \pi_n \qquad D_k^L$

each block is decreasing

Some notes on Sagan Ch 3:

Sec 3.1 (p. 94 Sagan)

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 3 & 6 & 5 & 1 & 7 \end{pmatrix} \xrightarrow[\text{(row)}]{\text{R-S}} \quad P = \begin{array}{|cccc|} 1 & 3 & 5 & 7 \\ 2 & 6 & & \\ 4 & & & \end{array} \qquad Q = \begin{array}{|cccc|} 1 & 3 & 4 & 7 \\ 2 & 5 & & \\ 6 & & & \end{array}$$

sage: RSK([4,2,3,6,5,1,7])

(row) insertion tableau

(row) recording tableau

Sec 3.4 (Sagan)

Def Say $\pi, \sigma \in S_n$ are P-equivalent

if $P(\pi) = P(\sigma)$.

E.g. $\pi = 2143$

$P(\pi)$	$Q(\pi)$
2	1
$\begin{array}{cc} 1 & 4 \\ 2 & \end{array}$	$\begin{array}{cc} 1 & 3 \\ 2 & \end{array}$
$\boxed{\begin{array}{c\|c} 1 & 3 \\ \hline 2 & 4 \end{array}}$	$\begin{array}{cc} 1 & 3 \\ 2 & 4 \end{array}$

2143 $\overset{K1}{\text{———}}$ 2413

K2

The equivalence class for $P = \begin{array}{|cc|} 1 & 3 \\ 2 & 4 \end{array}$ has these two permutations.

This makes sense because RS map is a bijection between S_n and pairs (P, Q) of standard tableaux.

If $P = \begin{array}{|c|c|} \hline 1 & 3 \\ \hline 2 & 4 \\ \hline \end{array}$, you can only pair P with $Q = \begin{array}{|c|c|} \hline 1 & 3 \\ \hline 2 & 4 \\ \hline \end{array}$ or

$Q = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$

Def

- Say $\pi \overset{K1}{\cong} \sigma$ if you can exchange

$$\underset{\text{mid}}{\underset{\uparrow}{y}}\ \underset{\text{smallest}}{\underset{\uparrow}{x}}\overset{\text{largest}}{z} \longleftrightarrow \underset{\text{mid}}{\underset{\uparrow}{y}}\ \overset{\text{largest}}{z}\underset{\text{smallest}}{\underset{\uparrow}{x}}$$

- Say $\pi \overset{K2}{\cong} \sigma$ if you can exchange

$$x\,z\,\underset{\text{middle}}{\underset{\uparrow}{y}} \longleftrightarrow z\,x\,\underset{\text{middle}}{\underset{\uparrow}{y}}$$

Def $\sigma \overset{K}{\cong} \pi$ if there is a sequence of Knuth moves to go from σ to π.

Thm $\pi \overset{K}{\cong} \sigma$ iff

$P(\pi) =$

Def Given a tableau $P = [R_1, R_2, \ldots, R_\ell]$, the row word (or row reading word, or reading word) of P is the permutation $\pi_P = R_\ell R_{\ell-1} \cdots R_1$

Eg If $P = \begin{array}{|c|c|c|c|}\hline 1 & 3 & 5 & 7 \\ \hline 2 & 6 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}$, then the row word of P is $\pi_P = 4\ 2\ 6\ 1\ 3\ 5\ 7$

If $P = \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 4 \\ \hline\end{array}$, then the row word of P is $\pi_P = 2\ 4\ 1\ 3$

Lemma Applying RS map to the row word π_P for tableau P would result in the row insertion tableau which is P.

Eg

$$4 \qquad \longleftarrow \qquad \underbrace{4}\ \underbrace{2\ 6}\ \underbrace{1\ 3\ 5\ 7}$$

$$\begin{array}{ll} 2 & 6 \\ 4 \end{array}$$

$$\begin{array}{ll} 1 & 6 \\ 2 \\ 4 \end{array}$$

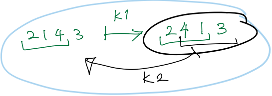
$\begin{array}{|c|c|c|c|}\hline 1 & 3 & 5 & 7 \\ \hline 2 & 6 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}$ insertion tableau for $4\,2\,6\,1\,3\,5\,7$

Practice Prove lemma 3.4.5 of Sagan Sec 3.4

1.9 "Fukuda" BBS convention and RSK conjectures

The "Fukuda" convention of BBS ("opp" to the first convention from notes Problem I, part 1)

Def Starting from an initial configuration,
a BBS move is to let each ball jump to the
next available spot (to the right),
starting with 1, 2, 3, ...



t=0 2 1 4 3
t=1 2 4 1 3
t=2 2 4 1 3

~~Content of the~~ soliton decomposition
~~Soliton:~~

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 4 \\ \cline{1-1} 2 \\ \cline{1-1}\end{array}$$

→ not a standard tableau

t=0 2 4 1 3
t=1 2 4 1 3
t=2 2 4 1 3

$$\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & 4 \\ \hline\end{array}$$

→ It is a standard tableau

Def The ~~Content~~ Soliton decomposition is the list of (increasing) blocks, where each block moves at the same speed (the length of the block) for each t (after t ⩾ large enough number)

t=0 4 2 3 6 | 5 1 7
t=1 4 6 2 1 3 5 7
t=2 4 6 2 1 3 5 7

⟷

$$\begin{array}{|c|c|c|c|}\hline 1 & 3 & 5 & 7 \\ \hline 2 \\ \cline{1-1} 6 \\ \cline{1-1} 4 \\ \cline{1-1}\end{array}$$

soliton decomposition ~~Content of Soliton~~ for

4 2 3 6 5 1 7

t=0 4 2 6 1 3 5 7
t=1 4 2 6 1 3 5 7

$$\begin{array}{|c|c|c|c|}\hline 1 & 3 & 5 & 7 \\ \hline 2 & 6 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}$$

soliton decomposition ~~Content of Soliton~~ for

$\pi = 4\,2\,6\,1\,3\,5\,7$ is the same as the (row) insertion tableau for π.

Def A descent in a permutation $\pi \in S_n$ is $i \in [n]$ where $\pi_i > \pi_{i+1}$

Sage: Permutation([. . .]).descents()

Notation (tentative / Think about good notations)

- Let $\textcircled{i}(\pi)$ = the length of a longest increasing subsequence.

  $I(\pi)$
- Let $ND(\pi) := 1 + |\text{descents of } \pi|$.

Let $R_K(\pi) = \max_{\pi = \underbrace{u_1 | u_2 | \dots | u_K}_{\text{consecutive}}} \left(i(u_1) + \dots + i(u_K) \right)$

Let $C_K(\pi) = \max_{\pi = u_1 \sqcup u_2 \dots \sqcup u_K} \left(ND(u_1) + \dots + ND(u_K) \right)$

"swap ascent w/ descent, decreasing w/ increasing"

Do not use this notation.

Use the notation in notes "Problem 1 part 3b"

Corollary Sec 1.9 (of "Localized Green's Thm" (Lemma 2.1 in LLPS) (from Notes Problem I, part 1))

Let soliton decomposition (π) be as described above.

ⓘ The ~~Soliton~~ soliton decomposition shape is a partition.

ⓘⓘ The shape of soliton decomposition (π) is given by SC shape = $(R_1, R_2 - R_1, R_3 - R_2, \dots)$.

ⓘⓘⓘ The conjugate of soliton decomposition shape is given by $(C_1, C_2 - C_1, C_3 - C_2, \dots)$.

REU Exercise 5 • Prove the corollary. • Think of notation you like.

REU Exercise 6 → See REU Report Overleaf

• Let T be a Standard tableau (increasing along rows, cols, has distinct entries $\{1, 2, \dots, n\}$).

6a (In REU Report + Week 6 Tue presentation)

- If π is the "forward" row reading word of T, prove that "Forward" Soliton Dec (π) = T.
- If π is the row reading word of T, prove that we see the soliton at $t=0$.
- As we move time forward, the configuration stays in the same order as π.
- If the system decomposes into solitons at $t=0$, then π is the row reading word of the row insertion P-tableau.

6b

- If π is the "backward row reading word" of T, prove "Backward" Sol Dec (π) = T.
- π is the "backward row reading word" of T iff the system decomposes into "backward" solitons at $t=0$

– Pick a convention for "backward tableau" (maybe convention #2)

	2
1	6
3	7
4	8
5	9

– If convention #2, modify the "backward RSK" to fit convention #2

Aubrey will work on this (July 6)

Conjectures (Problem I, Part 3):

① Soliton Shape (π) = RS_partition (π) iff soliton decomposition (π) = RS row insertion tableau] checked up to $n=12$

$\left(\underset{\text{conj}}{\Longrightarrow} \right)$

0 step:

Let $\pi \in S_n$, $P = P(\pi)$, r the row reading word for P.

Compute examples where $\kappa_1^{\pm}(\pi)$ when possible.

$\kappa_2^{\pm}(\pi)$

First step:

Let $\pi \in S_n$, $P = P(\pi)$, r the row reading word for P.

If $\kappa(\pi) = r$ where κ is one Knuth move that is not both κ_1, κ_2,

then applying one box ball move to π gives a config P.

②' Soliton Dec(π) is a Standard Young Tableau iff Soliton Dec(π) = $P(\pi)$.

↑(In REU Report I) Proposition

[Done ②' $\Longleftarrow$]

② Soliton Shape (π) = RS_partition (π) iff soliton decomposition (π) is a Standard Young tableau. } ($\Rightarrow$) is same as part ① ($\Leftarrow$) checked up to $n=11$

e.g. Soliton Content $\begin{pmatrix} 4 & 2 & 3 & 6 & 5 & 1 & 7 \end{pmatrix}$ is not a Standard Young tableau.

Still a conjecture ②:

$\lambda^{BBS}(\pi)$ $\lambda^{RS}(\pi)$

② If Soliton Shape (π) = RS_partition (π)

then Soliton Dec (π) is a Standard Young tableau.

③ If $\pi \in S_n$ avoids 2143 and avoids 3142, then
Soliton decomposition ~~Content~~ (π) is equal to the (row) RSK insertion tableau, the P-tableau } Checked up to n = 11
(Note: the other direction is false for $n \geq 5$)

④ If soliton shape of π is not equal λ, the shape of the RSK tableau,
then π contains 2143 as a pattern OR
π contains 3142 as a pattern.
(Note: the other direction is false for $n \geq 5$)
} A lot more permutations to check
checked up to n = 10

this statement is the contrapositive of conjecture ③

Notes on Conj ①

Lemma Each $R_i = A_i$ for all $i \in [n]$
iff Soliton shape $(\pi) = \lambda(\pi)$.

By REU Exercise 6,
Conjecture ① is true for row reading word and also all the permutations you get from applying BBS moves backward starting from a row reading word.

⑤ Problem: Give a criterion (using $\overset{\text{maybe}}{}$ pattern avoidance + others) for permutations $\pi \in S_n$ where Soliton Shape (π) = RS_partition (π).

Prop Sec 1.9 (Corollary of LLPS paper) One $\overset{\text{Fukuda}}{\text{BBS}}$ move is the same as applying F_1, n times.

Applying F_1 means :

- Make ball ① jump to the first available empty cell.
- Change ① to ⓝ
- Then decrease the value of all the other balls by 1.

E.g. $\pi =$ 2 1 4 3

$\xrightarrow{F_1}$ 1 _ 3 2 ④

$\xrightarrow{F_1}$ ④ 2 1 3

$\xrightarrow{F_1}$ 3 1 _ 2 ④

$\xrightarrow{F_1}$ $\boxed{2\ \_\ 4\ 1\ 3}$ is the configuration after doing one (Fukuda) BBS move.

—— end of lecture (week 2) Wed, June 3, 2020 ——

1.10 2143 and 3142 Pattern Avoidance

Ref: Wikipedia "Permutation Pattern" (Started by Knuth 1968)

Def Let $\pi \in S_n$.

- We say that π avoids the pattern $\underset{b\,a\,d\,c}{2143}$ (or π is 2143-avoiding) if (the 1-line notation of) π contains no subsequence b, a, d, c where $a < b < c < d$.

- Say π is $\underset{c\,a\,d\,b}{3142}$-avoiding if π contains no subsequence c, a, d, b where $a < b < c < d$.

- Say π contains 2143 as a pattern if π has a subsequence b, a, d, c where $a < b < c < d$.

- Say π contains 3142 as a pattern if π has a subsequence with the same relative order as 3142.

REU Exercise 7 (Use Fukuda BBS convention)

Let $n \leq 4$. Show $\pi \in S_n$ is 2143 avoiding and 3142 avoiding

if and only if

the soliton shape of π is equal to $\lambda(\pi)$, the shape of the RS partition

sage: Permutation([1,2,3,4]).RS_partition()

if and only if

the soliton content of π is equal to the (row) RSK insertion tableau, the P-tableau.

sage: RSK([1,2,3,4])[0]

More Conjectures (listed w/ the rest of PROBLEM I, part 3 conjectures above)

③ If $\pi \in S_n$ avoids 2143 and avoids 3142, then SolitonContent(π) is equal to the (row) RSK insertion tableau, the P-tableau

(Note: the other direction is false for $n \geq 5$)

~~④~~ If soliton shape of π is not equal λ, the shape of the RSK tableau, then π contains 2143 as a pattern OR π contains 3142 as a pattern.
(Note: the other direction is false for $n \geq 5$)

this statement is the contrapositive of conjecture ③

Correction June 8

– There are five perms in S_5 which have 2143 or 3142 as a pattern, but the same soliton ~~contents~~ decompositions as the P-tableaux.

$w_1 = 2\,5\,1\,4\,3$??

$w_2 = \begin{pmatrix}1&2&3&4&5\\3&5&1&4&2\end{pmatrix}$ is its own inverse

- 3 2 5 1 4
- 4 2 5 1 3

Two are row words of some (standard) tableaux

$\pi = 4\,1\,3\,5\,2 \xrightarrow{RSK} \begin{array}{lll}1&2&5\\3&&\\4&&\end{array}\ ,\ \begin{array}{lll}1&3&4\\2&&\\5&&\end{array}$

Note: $\pi^{-1} = \pi^{\text{reverse}}$
$= \pi^{\text{complement}}$

not a row word,
not a "backward row" word

– There are 76, 774, 6821 for $n=6$, $n=7$, $n=8$

$W_1 = 25\ 143$

$W_2\ \ 35\ 142$

are both "backward row" words, for $\begin{array}{|cc|}\hline 2 & 5 \\ 1 & 4 \\ & 3 \\ \hline\end{array}$ $\begin{array}{|cc|}\hline 5 & 2 \\ 4 & 1 \\ 3 & \\ \hline\end{array}$

$\begin{array}{|cc|}\hline 3 & 5 \\ 1 & 4 \\ & 2 \\ \hline\end{array}$ $\begin{array}{|cc|}\hline 5 & 3 \\ 4 & 1 \\ 2 & \\ \hline\end{array}$

$W_2 = 35142 \xrightarrow[\text{RS"}]{\text{"backward}}$

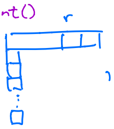
P	Q
2	5
4 2	5 4
14 2	35 4
15 4 2	35 4 2
$\begin{array}{\|cc\|}\hline 3 & 5 \\ 1 & 4 \\ & 2 \\ \hline\end{array}$	$\begin{array}{\|cc\|}\hline 3 & 5 \\ 1 & 4 \\ & 2 \\ \hline\end{array}$

Def The complement of a permutation $\pi_1 \pi_2 \cdots \pi_n$ is $n+1-\pi_1, n+1-\pi_2, \ldots, n+1-\pi_n$.

`sage: pi.complement()`

Task: Make sure you believe this

Rem If $\lambda(\pi) = P(\pi)$ is of $(r, 1, 1, 1, \ldots)$



, then the soliton shape must be $\lambda(\pi)$.

⑤ Question: Give a criterion (using maybe pattern avoidance + others) for permutations $\pi \in S_n$ where Soliton Shape (π) = RS_partition (π).

Sage practice for checking if π is a row word (reading word)

```
sage: RSK([4,1,3,5,2])[0].reading_word_permutation() ==
        Permutation([4,1,3,5,2])
False
sage: RSK([5,2,1,3,4])[0].reading_word_permutation() ==
        Permutation([5,2,1,3,4])
True
```

False ← means π is not a row word

True ← means π is a row word

Sage Practice for pattern avoidance

```
sage: w=Permutation([6,2,1,3,7,5,4])
sage: w.avoids([2,1,4,3])    # should be False because 2,1,7,5 is a subseq
False
sage: for n in range(2,10):
          P = Permutations(n, avoiding=[[2,1,4,3], [3,1,4,2]])    # 2143 avoiding AND 3142 avoiding
          print(n, len(P))
```

(2,1,7,5 is a subseq: 2 1 4 3)

n	len(P)
2	2
3	6
4	22
5	90
6	395
7	1823
8	8741
9	43193

} all permutations avoid 2143 and 3142 (n = 2, 3)

← all permutations except 2143 and 3142 (n = 4)

← all but 30 permutations (n = 5)

} all but factorial(n) − len(P) factorials (n = 6, …, 9)

OEIS task

- Go to oeis.org (the Online Encyclopedia of Integer Sequences) and enter 2, 6, 22, 90, 395, 1823, 8741, 43193 in the search bar.
  This particular sequence would match with an entry.
- Try to look up other sequences you see this summer (when running computation)

1.11 Biword, dual biword, P-tableau as conserved quantity

(Ref: Fukuda Sec 3)

Def (Biword and dual biword)

- Each BBS configuration with balls colored $[n]$ can be described as a biword (or generalized permutation)

$$w = \begin{pmatrix} i_1 & i_2 & \cdots & i_k & \cdots & i_n \\ w_{i_1} & w_{i_2} & \cdots & w_{i_k} & \cdots & w_{i_n} \end{pmatrix}$$

where $i_k < i_{k+1}$, and the box at position i_k contains the ball $\textcircled{w_{i_k}}$, for all k.

Symbol 0 denotes an empty cell

E.g. Configuration

$$\begin{array}{cccccccccccccc} & \text{box } 1 & \text{box } 2 & \text{box } 3 & \text{box } 4 & \text{box } 5 & \text{box } 6 & \text{box } 7 & \text{box } 8 & \text{box } 9 & \text{box } 10 & \text{box } 11 & \text{box } 12 & \\ \cdots & 0 & \textcircled{4} & 0 & 0 & \textcircled{6} & \textcircled{2} & 0 & \textcircled{1} & \textcircled{3} & \textcircled{5} & \textcircled{7} & \cdots & \end{array}$$

is represented by the biword

$$w = \begin{pmatrix} 2 & 5 & 6 & 8 & 9, & 10, & 11 \\ 4 & 6 & 2 & 1 & 3 & 5 & 7 \end{pmatrix}$$

Configuration

$$\begin{array}{ccccccccccccc} & \text{box } 1 & \text{box } 2 & \text{box } 3 & \text{box } 4 & \text{box } 5 & \text{box } 6 & \text{box } 7 & \text{box } 8 & \text{box } 9 & \text{box } 10 & \text{box } 11 & \text{box } 12 \\ \cdots & \textcircled{4} & \textcircled{2} & \textcircled{3} & \textcircled{6} & \textcircled{5} & \textcircled{1} & \textcircled{7} & 0 & 0 & 0 & 0 & 0 \cdots \end{array}$$

is represented by the biword

$$w = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 3 & 6 & 5 & 1 & 7 \end{pmatrix}$$

- Given a biword $w = \begin{pmatrix} i_1 & i_2 & \cdots & i_k & \cdots & i_n \\ w_{i_1} & w_{i_2} & \cdots & w_{i_k} & \cdots & w_{i_r} \end{pmatrix}$, its dual biword w^* is of the form

$$w^* = \begin{pmatrix} 1 & 2 & \cdots & k & \cdots & n \\ b_1 & b_2 & \cdots & b_k & \cdots & b_n \end{pmatrix}$$

where b_k is the position of the box which contains the ball $\textcircled{k}$.

E.g. the dual biword of $w = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 3 & 6 & 5 & 1 & 7 \end{pmatrix}$ is $w^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 2 & 3 & 1 & 5 & 4 & 7 \end{pmatrix}$.

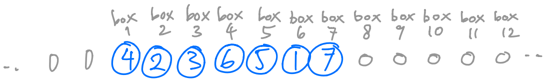
Note If w is a permutation in 2-line notation, then w^* is the 2-line notation of w^{-1}, the inverse of w (as a function)

E.g. the dual biword of $w = \begin{pmatrix} 2 & 5 & 6 & 8 & 9, & 10, & 11 \\ 4 & 6 & 2 & 1 & 3 & 5 & 7 \end{pmatrix}$ is $w^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 8 & 6 & 9 & 2, & 10, & 5, & 11 \end{pmatrix}$.

- In [Fukuda], the bottom row of the dual biword w^* of a configuration c. is called the box-label sequence associated with the config c.

E.g. Same example as above

Configuration



biword

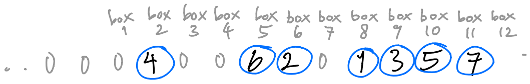
$$w = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 3 & 6 & 5 & 1 & 7 \end{pmatrix}$$

dual biword

$$w^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 2 & 3 & 1 & 5 & 4 & 7 \end{pmatrix}.$$

↳ after one (Fukuda) BBS move

box-ball sequence $b = (6, 2, 3, 1, 5, 4, 7)$



$$w' = \begin{pmatrix} 2 & 5 & 6 & 8 & 9, & 10, & 11 \\ 4 & 6 & 2 & 1 & 3 & 5 & 7 \end{pmatrix} \quad (w')^* = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 8 & 6 & 9 & 2, & 10, & 5, & 11 \end{pmatrix}.$$

box-ball sequence $b' = (8, 6, 9, 2, 10, 5, 11)$

Given a BBS configuration c, we associate to c a pair (P, Q) of tableaux by applying the RSK algorithm to the biword w. (see Sec 2.4 of [Fukuda] for RSK on a biword)

$$w = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 3 & 6 & 5 & 1 & 7 \end{pmatrix} \xrightarrow{RSK}$$

P	Q
$\begin{matrix} 2 & 3 & 6 \\ 4 \end{matrix}$	$\begin{matrix} 1 & 3 & 4 \\ 2 \end{matrix}$
$\begin{matrix} 2 & 3 & 5 \\ 4 & 6 \end{matrix}$	$\begin{matrix} 1 & 3 & 4 \\ 2 & 5 \end{matrix}$

$$P = \begin{array}{|cccc} 1 & 3 & 5 & 7 \\ 2 & 6 \\ 4 \end{array} \qquad Q = \begin{array}{|cccc} 1 & 3 & 4 & 7 \\ 2 & 5 \\ 6 \end{array}$$

insertion tableau of w — recording tableau

$$w' = \begin{pmatrix} 2 & 5 & 6 & 8 & 9, & 10, & 11 \\ 4 & 6 & 2 & 1 & 3 & 5 & 7 \end{pmatrix} \xrightarrow{RSK}$$

P	Q
4	2
$\begin{matrix} 4 & 6 \end{matrix}$	$\begin{matrix} 2 & 5 \end{matrix}$
$\begin{matrix} 2 & 6 \\ 4 \end{matrix}$	$\begin{matrix} 2 & 5 \\ 6 \end{matrix}$
$\begin{matrix} 1 & 6 \\ 2 \\ 4 \end{matrix}$	$\begin{matrix} 2 & 5 \\ 6 \\ 8 \end{matrix}$

Sage Math

```
Sage: Wtop = [2,5,6,8,9,10,11]
sage : W bottom = [4,6,2,1,3,5,7]
sage : RSK(Wtop, Wbottom)
```

$$P = \begin{array}{|cccc} 1 & 3 & 5 & 7 \\ 2 & 6 \\ 4 \end{array} \qquad Q = \begin{array}{|cccc} 2 & 5, & 10, & 11 \\ 6 & 9 \\ 8 \end{array}$$

[Thm 3.6.6, Sagan Sec 3.6] If $\pi \in S_n$, $P(\pi^{-1}) = Q(\pi)$ and $Q(\pi^{-1}) = P(\pi)$

[A more general version] If w is a biword, $P(w^*) = Q(w)$ and $Q(w^*) = P(w)$

i.e. the P-tableau of a biword w is achieved by RSK row-incertion of the bottom line of w, by def of RSK. The Q-tableau of a biword w can be achieved by RSK row-insertion of the bottom line of the dual biword w^*. by

[Thm 3.1, Fukuda]

Given a BBS (a list of infinite configurations $c_0, c_1, c_2, \dots$ starting from time $t=0$, then apply one BBS move for each time increment), associate each configuration c_t with a pair of tableaux (P, Q) as explained above.

1. The P-tableau is a conservsed quantity under the time evolution of the BBS.

   That is, let w_0 be the biword which represents the configuration at time $t=0$, and let P_0 be the insertion tableau of w_0. Then, for each time t, the insertion tableau of the biword w_t which represents the configuration c_t at time t is equal to P_0.

2. The Q-tableau evolves independently of the P-tableau.

Proofs are given in [Sec 4, Fukuda] using concepts in [Sec 3, Fukuda].

end Thurs, June 4, 2020

Possible task

Study the proofs in [Sec 4, Fukuda] and concepts from [Sec 3, Fukuda]

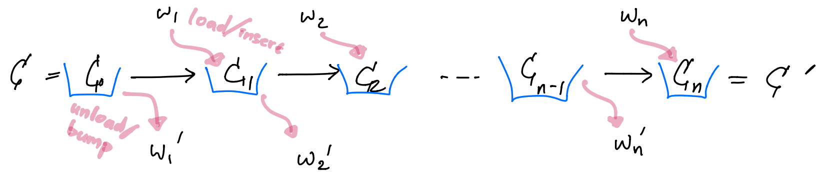
1.12 Carrier algorithm

(Ref: Fukuda Sec 3.3 "Carrier algorithm")
and Sec 3.4 "Time evolution w/ a carrier")

The carrier algorithm is a way to transform a finite sequence $w = (w_1, w_2, \dots, w_n)$ into another sequence $w' = (w_1', w_2', \dots, w_n')$ using a weakly increasing sequence $\mathcal{C} = (c_1, c_2, \dots, c_m)$, called the carrier.

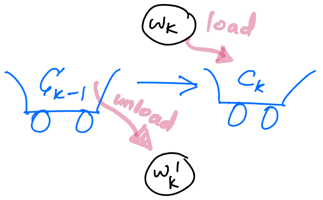
The carrier moves along w from left to right,

while the carrier passes a number w_k,
the carrier loads w_k and unloads w_k'.



Rule of loading and unloading for the carrier :

- Say the carrier is about to pass the number w_k in the sequence w, So we are about to load w_k onto the carrier.



- Let $\mathcal{C}_{k-1} = (c_1, c_2, \dots, c_m)$ be the sequence of numbers which have been loaded on the carrier.
  $c_1 \leq c_2 \leq \cdots \leq c_m$

- Compare w_k with the numbers in the carrier $\mathcal{C}_{k-1}$.

→ If there are numbers larger than w_k in the carrier $\mathcal{C}_{k-1}$, then unload a smallest of those numbers, and load w_k in its place, into the carrier.

→ Otherwise, all the numbers in the carrier $\mathcal{C}_{k-1}$ are smaller/equal to w_k. Then unload a minimum entry in the carrier $\mathcal{C}_{k-1}$, and load w_k onto the carrier.

More precisely,

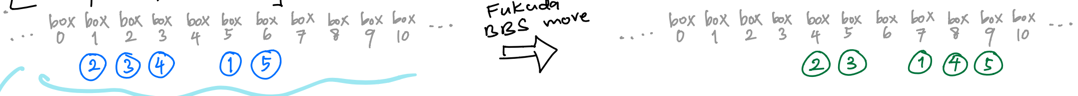
$$w_k' = \begin{cases} \min\{\, c_i \in \mathcal{C}_{k-1} \mid w_k < c_i \,\} & \text{if there are numbers in the carrier } \mathcal{C}_{k-1} \text{ larger than } w_k, \\ c_1 & \text{otherwise.} \end{cases}$$

$\mathcal{C}_k$ is the sequence we get from the previous carrier by unloading w_k' and loading w_k.

Def Given sequences $\mathcal{C} = (c_1, c_2, \dots, c_m)$ and $w = (w_1, w_2, \dots, w_n)$,
$c_1 \leq c_2 \leq \cdots \leq c_m$
start with $\mathcal{C}_0 := \mathcal{C}$ and repeat the rule of unloading/loading for each $k = 1, \dots, n$ to get new sequences $\mathcal{C}' := \mathcal{C}_n$ and w'.
Call the transformation $(\mathcal{C}, w) \longrightarrow (\mathcal{C}', w')$ the carrier algorithm.

1.12.(i) "Original" Carrier algorithm using the configuration

[Example 4, Fukuda]

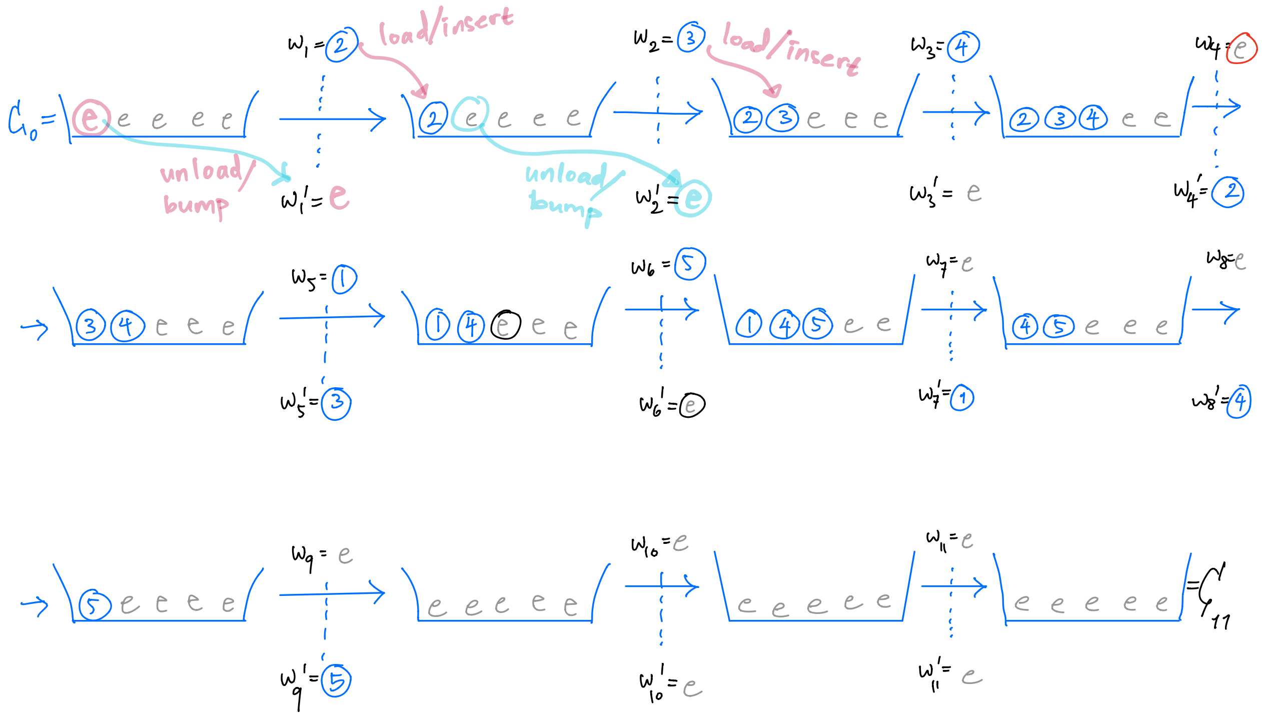


[Example 6, Fukuda] "Original" Carrier algorithm using the configuration

$W := \text{Config} = (②,③,④,e,①,⑤,e,e,e,e,e)$ and the carrier $C = (\overbrace{e,e,e,e,e}^{n = \#\text{ balls}})$

$w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7\ w_8\ w_9\ w_{10}\ w_{11}$

Here let e denote an empty cell, and think of $e := n+1. = 6$



The carrier algorithm gives us $w' = (e,e,e,2,3,e,1,4,5,e,e)$.

[Prop 3.2, Fukuda] "Original" Carrier algorithm using the configuration

One (Fukuda) BBS move can be described by the carrier algorithm,

with $C = \underbrace{(e, e, \dots, e)}_{n}$ as the initial state of the carrier

The final state of the carrier will again be $(\overbrace{e,e,\dots,e}^{n})$. ⊏

1.12.(ii) Carrier algorithm using the box-ball sequence

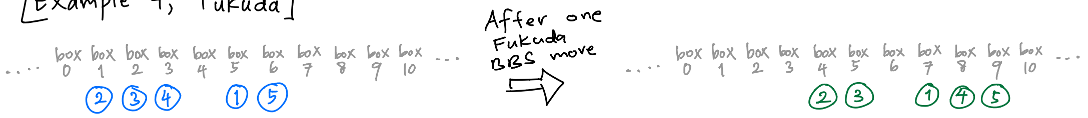
Recall the box-label sequence $b = (b_1, b_2, \ldots, b_k, \ldots, b_n)$, the bottom row of the dual biword w^*. (See [Sec 3.2, Fukuda].)

[Prop 3.3, Fukuda] Carrier algorithm using the box-ball sequence

One (Fukuda) BBS move, as a transformation of the box-label sequence $b \to b'$ can be described by the carrier algorithm w/ the initial state of the carrier $C' = (\ell_1, \ell_2, \ldots, \ell m)$ defined as the increasing sequence consisting of the positions/labels of all empty boxes in the interval $[p, q]$. □

p, q are chosen so that all the balls of the two configurations are between box p and box q.

[Example 4, Fukuda]



Dual biword $w^* = \begin{pmatrix} ① & ② & ③ & ④ & ⑤ \\ 5 & 1 & 2 & 3 & 6 \end{pmatrix}$

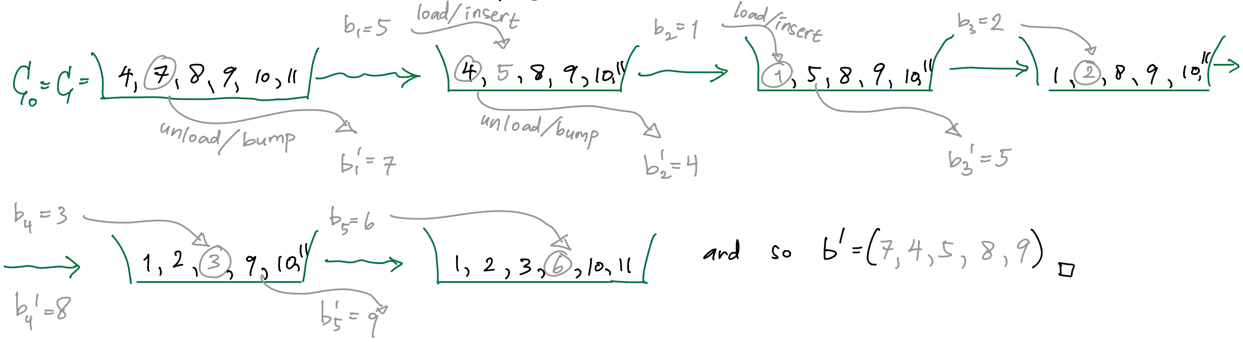
New dual biword $\begin{pmatrix} ① & ② & ③ & ④ & ⑤ \\ 7 & 4 & 5 & 8 & 9 \end{pmatrix}$ □

[Example 7, Fukuda] Carrier algorithm using the box-ball sequence

Take $[p, q] = [1, 11]$

Dual biword $w^* = \begin{pmatrix} ① & ② & ③ & ④ & ⑤ \\ 5 & 1 & 2 & 3 & 6 \end{pmatrix}$, so the box-ball sequence is $b = (5, 1, 2, 3, 6)$, the bottom row of the dual biword w^*.

The initial configuration is $C' = (4, 7, 8, 9, 10, 11)$, the labels of all the empty boxes in the interval $[1, 11]$.



and so $b' = (7, 4, 5, 8, 9)$ □

1.13 Carrier algorithm: bumping in RSK and Knuth transformation

Remark, also stated in [Sec 2.2, Fukuda], [pg. 102, Sec 3.4, Sagan]

The bumping algorithm in the RS correspondence can be described as a move

(i) $(c \ d_1 d_2 \dots)\, b \xmapsto[\text{bump } c]{\text{insert } b} c \ (b \ d_1 d_2 \dots)$

or

(ii) $(a_1 a_2 \dots a_r c)\, b \xmapsto[\text{bump } c]{\text{insert } b} c\,(a_1 a_2 \dots a_r \ b)$

or

(iii) $(a_1 a_2 \dots a_r \, c \, d_1 d_2 \dots)\, b \xmapsto[\text{bump } c]{\text{insert } b} c\,(a_1 a_2 \dots a_r b \ d_1 d_2 \dots)$

Examples:

$\overset{\text{"c"}}{7} \ 9, 11, 20 \xmapsto[\text{bump } 7]{\text{insert } 5} \overset{\text{"b"}}{5} \ 9, 11, 20$

$2 \ 3 \ \overset{\text{"c"}}{7} \xmapsto[\text{bump } 7]{\text{insert } 5} 2 \ 3 \ \overset{\text{"b"}}{5}$

$2 \ 3 \ \overset{\text{"c"}}{7} \ 9, 11, 20 \xmapsto[\text{bump } 7]{\text{insert } 5} 2 \ 3 \ \overset{\text{"b"}}{5} \ 9, 11, 20$

for $a_1 a_2 \dots a_r \leq b < c \leq d_1 \leq d_2 \dots$

E.g: $c d_1 d_2 \ b \overset{(1)}{\mapsto} c \ d_1 b d_2 \overset{(1)}{\mapsto} c \ b \, d_1 d_2$

- Situation (i) shows two words which differ by a sequence of Knuth transformations of the first kind (See Sagan Sec 3.4)
- Situation (ii) shows two words which differ by a sequence of Knuth transformations of the second kind.
- The transformation (iii) is achieved by applying (i) then (ii)

E.g: $a \ \underline{c \ d \ b} \overset{(i)}{\longmapsto} a \ \underline{c \ b \ d} = \overbrace{a \ c \ b} \ d \overset{(ii)}{\longmapsto} \overbrace{c \ a \ b} \ d$

(iv) Def A trivial transformation is when no letters are rearranged:

$(x \ b_1 b_2 \dots b_r)\, z \xmapsto[\text{bump } x]{\text{insert } z} x\,(b_1 b_2 \dots b_r \ z)$

e.g. $(4 \ 5 \ 7 \ 8 \ 8)\, 8 \xmapsto[\text{bump } 4]{\text{insert } 8} 4\,(5 \ 7 \ 8 \ 8 \ 8)$

where $x \leq b_1 \leq b_2 \leq \dots \leq b_r \leq z$

Note We don't do (iv) in the RS algorithm.
But we do

$(x \ b_1 b_2 \dots b_r)\, z \xmapsto[\text{no bumping}]{\text{insert } z} (x \ b_1 b_2 \dots b_r \ z)$

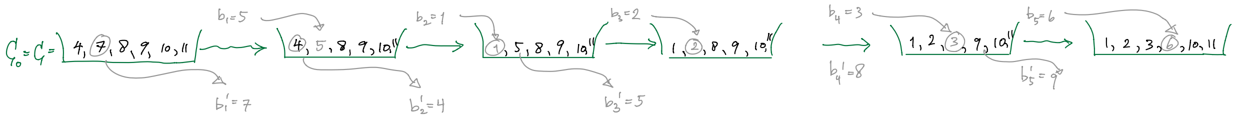
[Remark 4, Fukuda Sec 3.3]

Each step of loading / unloading is applying rule (iii) (sequence of two Knuth transformations)
inserting / bumping see previous page
or the trivial transformation (iv)

▣ An example of [Remark 4, Fukuda]

From [Example 7, Fukuda] Carrier algorithm using the box-ball sequence

$b = (5, 1, 2, 3, 6)$



$(4\ 7\ 8)\,5 \xrightarrow[\text{bump } 7]{\text{insert } 5} 7\,(4\,5\,8)$ $\quad (45)\,1 \xrightarrow[\text{bump } 4]{\text{insert } 1} 4\,(15)$ $\quad (158)\,2 \xrightarrow[\text{bump } 5]{\text{insert } 2} 5\,(128)$ $\quad (289)\,3 \xrightarrow[\text{bump } 8]{\text{insert } 3} 8\,(239)$ $\quad (3,9,10)\,6 \xrightarrow[\text{bump } 9]{\text{insert } 6} 9\,(3,6,10)$

$4<5<7<8$ $\quad 1<4<5$

and so $b' = (7, 4, 5, 8, 9)$

$$\zeta b = \zeta_0 b_1 b_2 b_3 b_4 b_5 \longrightarrow b_1' \zeta_1' b_2 b_3 b_4 b_5 \longrightarrow b_1' b_2' \zeta_2' b_3 b_4 b_5 \longrightarrow b_1' b_2' b_3' \zeta_3' b_4 b_5 \longrightarrow b_1' b_2' b_3' b_4' \zeta_4' b_5 \longrightarrow b_1' b_2' b_3' b_4' b_5' \zeta_5' = b' \zeta_5$$

$$\zeta b = \zeta_0 b_1 b_2 b_3 b_4 b_5 = \left(\underbrace{4, 7, 8, 9, 10, 11}_{\zeta_0}\right) \underset{b_1}{5}\ \underset{b_2}{1}\ \underset{b_3}{2}\ \underset{b_4}{3}\ \underset{b_5}{6} \xrightarrow{(iii)}$$

$$b_1' \zeta_1' b_2 b_3 b_4 b_5 = \underset{b_1'}{7} \left(\underbrace{4, 5, 8, 9, 10, 11}_{\zeta_1}\right) \underset{b_2}{1}\ \underset{b_3}{2}\ \underset{b_4}{3}\ \underset{b_5}{6} \xrightarrow{(i)}$$

$$b_1' b_2' \zeta_2 b_3 b_4 b_5 = \underset{b_1'}{7}\ \underset{b_2'}{4} \left(\underbrace{1, 5, 8, 9, 10, 11}_{\zeta_2}\right) \underset{b_3}{2}\ \underset{b_4}{3}\ \underset{b_5}{6} \xrightarrow{(iii)}$$

$$b_1' b_2' b_3' \zeta_3 b_4 b_5 = \underset{b_1'}{7}\ \underset{b_2'}{4}\ \underset{b_3'}{5} \left(\underbrace{1, 2, 8, 9, 10, 11}_{\zeta_3}\right) \underset{b_4}{3}\ \underset{b_5}{6} \xrightarrow{(iii)}$$

$$b_1' b_2' b_3' b_4' \zeta_4 b_5 = \underset{b_1'}{7}\ \underset{b_2'}{4}\ \underset{b_3'}{5}\ \underset{b_4'}{8} \left(\underbrace{1, 2, 3, 9, 10, 11}_{\zeta_4}\right) \underset{b_5}{6} \xrightarrow{(iii)}$$

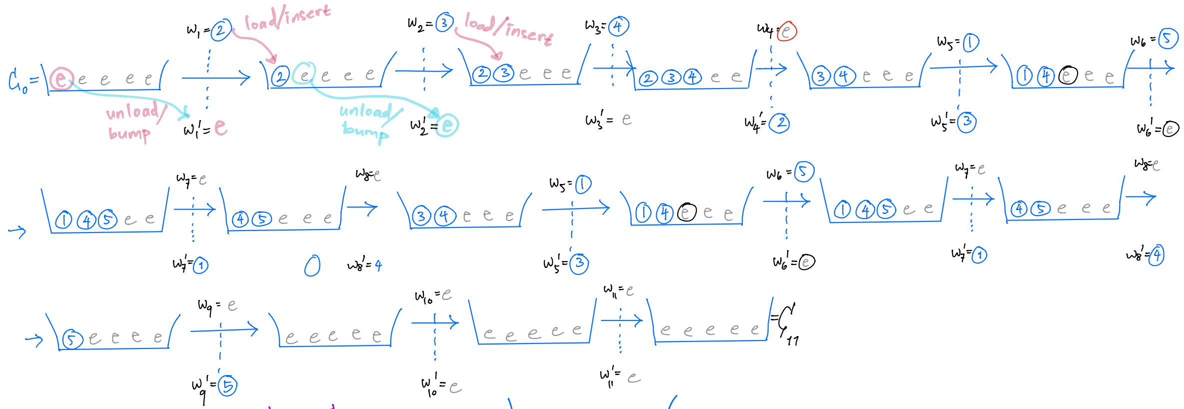
$$b_1' b_2' b_3' b_4' b_5' \zeta_5 = \underset{b_1'}{7}\ \underset{b_2'}{4}\ \underset{b_3'}{5}\ \underset{b_4'}{8}\ \underset{b_5'}{9} \left(\underbrace{1, 2, 3, 6, 10, 11}_{\zeta_5}\right) = b' \zeta_5$$

▫ Another example of [Remark 4, Fukuda]

From [Example 6, Fukuda] "Original" Carrier algorithm using the configuration

$w := \text{Config} = (\underset{w_1}{②}, \underset{w_2}{③}, \underset{w_3}{④}, \underset{w_4}{e}, \underset{w_5}{①}, \underset{w_6}{⑤}, \underset{w_7}{e}, \underset{w_8}{e}, \underset{w_9}{e}, \underset{w_{10}}{e}, \underset{w_{11}}{e})$ and the carrier $\zeta = (e, e, e, e, e)$

Here let e denote an empty cell, and think of $e := n+1. = 6$



$$\zeta w = \zeta_0 w_1 w_2 w_3 \dots w_{11} = \underbrace{\backslash\ e\ e\ e\ e\ e\ /}_{\zeta_0} \underset{w_1}{2}\ \underset{w_2}{3}\ \underset{w_3}{4}\ \underset{w_4}{e}\ \underset{w_5}{1}\ \underset{w_6}{5}\ \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{(i)}$$

$$w_1' \zeta_1 w_2 w_3 \dots w_{11} = \underset{w_1'}{e}\ \underbrace{\backslash\ 2\ e\ e\ e\ e\ /}_{\zeta_1} \underset{w_2}{3}\ \underset{w_3}{4}\ \underset{w_4}{e}\ \underset{w_5}{1}\ \underset{w_6}{5}\ \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{(iii)}$$

$$w_1' w_2' \zeta_2 w_3 \dots w_{11} = \underset{w_1'}{e}\ \underset{w_2'}{e}\ \underbrace{\backslash\ 2\ 3\ e\ e\ e\ /}_{\zeta_2} \underset{w_3}{4}\ \underset{w_4}{e}\ \underset{w_5}{1}\ \underset{w_6}{5}\ \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{(iii)}$$

$$w_1' w_2' w_3' \zeta_3 w_4 \dots w_{11} = \underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underbrace{\backslash\ 2\ 3\ 4\ e\ e\ /}_{\zeta_3} \underset{w_4}{e}\ \underset{w_5}{1}\ \underset{w_6}{5}\ \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{\text{trivial}} (iv)$$

$$w_1' w_2' w_3' w_4' \zeta_4 w_5 \dots w_{11} = \underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underbrace{\backslash\ 3\ 4\ e\ e\ e\ /}_{\zeta_4} \underset{w_5}{1}\ \underset{w_6}{5}\ \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{(i)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underbrace{\backslash\ 1\ 4\ e\ e\ e\ /}_{\zeta_5} \underset{w_6}{5}\ \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{(iii)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underset{w_6'}{e}\ \underbrace{\backslash\ 1\ 4\ 5\ e\ e\ /}_{\zeta_6} \underset{w_7}{e}\ \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{\text{trivial } (iv)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underset{w_6'}{e}\ \underset{w_7'}{1}\ \underbrace{\backslash\ 4\ 5\ e\ e\ e\ /}_{\zeta_7} \underset{w_8}{e}\ \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{\text{trivial } (iv)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underset{w_6'}{e}\ \underset{w_7'}{1}\ \underset{w_8'}{4}\ \underbrace{\backslash\ 5\ e\ e\ e\ e\ /}_{\zeta_8} \underset{w_9}{e}\ \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{\text{trivial } (iv)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underset{w_6'}{e}\ \underset{w_7'}{1}\ \underset{w_8'}{4}\ \underset{w_9'}{5}\ \underbrace{\backslash\ e\ e\ e\ e\ e\ /}_{\zeta_9} \underset{w_{10}}{e}\ \underset{w_{11}}{e} \quad \xrightarrow{\text{trivial } (iv)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underset{w_6'}{e}\ \underset{w_7'}{1}\ \underset{w_8'}{4}\ \underset{w_9'}{5}\ \underset{w_{10}'}{e}\ \underbrace{\backslash\ e\ e\ e\ e\ e\ /}_{\zeta_{10}} \underset{w_{11}}{e} \quad \xrightarrow{\text{trivial } (iv)}$$

$$\underset{w_1'}{e}\ \underset{w_2'}{e}\ \underset{w_3'}{e}\ \underset{w_4'}{2}\ \underset{w_5'}{3}\ \underset{w_6'}{e}\ \underset{w_7'}{1}\ \underset{w_8'}{4}\ \underset{w_9'}{5}\ \underset{w_{10}'}{e}\ \underset{w_{11}'}{e}\ \underbrace{\backslash\ e\ e\ e\ e\ e\ /}_{\zeta_{11}} = w_1' w_2' w_3' w_4' w_5' w_6' w_7' w_8' w_9' w_{10}' w_{11}' \zeta_{11} = w' \zeta_{11}$$

The carrier algorithm gives us $w' = (e, e, e, 2, 3, e, 1, 4, 5, e, e)$.

~ End Fri, June 5, 2020