Depth in classical Coxeter groups

Riccardo Biagioli (Lyon 1)

Eli Bagno, Mordechai Novick (Jerusalem College of Tech.)
and Alexander Woo (U. Idaho)

Journées nationales 2017 GDR-IM
Montpellier, 15 mars 2017

Preliminaries

S_n denotes the symmetric group on $[n] = \{1, \ldots, n\}$

Definition

Let $w \in S_n$. The entry $i \in [n]$ is an **exceedance** of w if $w(i) > i$. For an exceedance i of w, the **exceedance size** of i is $w(i) - i$. The entry $i \in [n]$ is a **descent** of w if $w(i) > w(i+1)$.

Consider

$$w = \begin{pmatrix} \boxed{1} & \boxed{2} & 3 & 4 & 5 \\ 2 & 5 & 1 & 4 & 3 \end{pmatrix} = [2, 5, 1, 4, 3] \in S_5.$$

The exceedances are 1 and 2. The exceedance sizes are $2 - 1 = 1$ and $5 - 2 = 3$. The descents of w are 2 and 4.

Preliminaries

S_n denotes the symmetric group on $[n] = \{1, \ldots, n\}$

Definition

Let $w \in S_n$. The entry $i \in [n]$ is an **exceedance** of w if $w(i) > i$. For an exceedance i of w, the **exceedance size** of i is $w(i) - i$. The entry $i \in [n]$ is a **descent** of w if $w(i) > w(i+1)$.

Consider

$$w = \begin{pmatrix} \boxed{1} & \boxed{2} & 3 & 4 & 5 \\ 2 & 5 & 1 & 4 & 3 \end{pmatrix} = [2, 5, 1, 4, 3] \in S_5.$$

The exceedances are 1 and 2. The exceedance sizes are $2 - 1 = 1$ and $5 - 2 = 3$. The descents of w are 2 and 4.

Sorting by transpositions

One can imagine various “machines” that can sort permutations (to the identity) by swapping pairs of entries.

Machine ℓ: Can only swap adjacent entries, and every move costs 1.

Machine a: Can swap arbitrary pairs of entries, and every move costs 1.

Machine d: Can swap arbitrary pairs of entries, and a move costs the distance between the entries.

Question

Can we look at a permutation and easily tell the minimum cost to sort it?

Inversions

Machine ℓ

For Machine ℓ, the answer is called the **length** of the permutation, and it is equal to the **number of inversions**:

$$inv(w) = |\{(i,j) \in [n] \times [n] \mid i < j \text{ and } w(i) > w(j)\}|.$$

One optimal algorithm is to always swap the rightmost descent.

For $w = 2431756$, we have

$$2431\mathbf{7}56 \xrightarrow{(56)} 24315\mathbf{7}6 \xrightarrow{(67)} 24\mathbf{3}1567 \xrightarrow{(34)} 2\mathbf{4}13567$$
$$\xrightarrow{(23)} 21\mathbf{4}3567 \xrightarrow{(34)} \mathbf{2}134567 \xrightarrow{(12)} 1234567$$

So $w = (12)(34)(23)(34)(67)(56)$, in fact w has 6 inversions.

Inversions

Machine ℓ

For Machine ℓ, the answer is called the **length** of the permutation, and it is equal to the **number of inversions**:

$$inv(w) = |\{(i,j) \in [n] \times [n] \mid i < j \text{ and } w(i) > w(j)\}|.$$

One optimal algorithm is to always swap the rightmost descent.

For $w = 2431756$, we have

$$2431\mathbf{7}56 \overset{(56)}{\to} 24315\mathbf{7}6 \overset{(67)}{\to} 24\mathbf{3}1567 \overset{(34)}{\to} 2\mathbf{4}13567$$
$$\overset{(23)}{\to} 21\mathbf{4}3567 \overset{(34)}{\to} \mathbf{2}134567 \overset{(12)}{\to} 1234567$$

So $w = (12)(34)(23)(34)(67)(56)$, in fact w has 6 inversions.

Inversions

Machine ℓ

For Machine ℓ, the answer is called the **length** of the permutation, and it is equal to the **number of inversions**:

$$inv(w) = |\{(i,j) \in [n] \times [n] \mid i < j \text{ and } w(i) > w(j)\}|.$$

One optimal algorithm is to always swap the rightmost descent.

For $w = 2431756$, we have

$$2431\mathbf{7}56 \overset{(56)}{\rightarrow} 24315\mathbf{7}6 \overset{(67)}{\rightarrow} 24\mathbf{3}1567 \overset{(34)}{\rightarrow} 2\mathbf{4}13567$$
$$\overset{(23)}{\rightarrow} 21\mathbf{4}3567 \overset{(34)}{\rightarrow} \mathbf{2}134567 \overset{(12)}{\rightarrow} 1234567$$

So $w = (12)(34)(23)(34)(67)(56)$, in fact w has 6 inversions.

Cycles

Machine a

For Machine a, the answer is called the **absolute length**, and it is equal to n **minus the number of cycles**.

One optimal algorithm (called **straigh selection sort** by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2431756$, we have

$$2431\mathbf{7}56 \overset{(57)}{\rightarrow} 2431\mathbf{6}57 \overset{(56)}{\rightarrow} 2\mathbf{4}31567 \overset{(24)}{\rightarrow} \mathbf{2}134567 \overset{(12)}{\rightarrow} 1234567$$

We obtain $w = 2431756 = (12)(24)(56)(57) = (124)(3)(576)$, and so $a(w) = 4 = n - \#\text{cycles}$.

Cycles

Machine *a*

For Machine *a*, the answer is called the **absolute length**, and it is equal to ***n* minus the number of cycles**.

One optimal algorithm (called **straigh selection sort** by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2431756$, we have

$$2431\mathbf{7}56 \overset{(57)}{\to} 2431\mathbf{6}57 \overset{(56)}{\to} 2\mathbf{4}31567 \overset{(24)}{\to} \mathbf{2}134567 \overset{(12)}{\to} 1234567$$

We obtain $w = 2431756 = (12)(24)(56)(57) = (124)(3)(576)$, and so $a(w) = 4 = n - \#\text{cycles}$.

Cycles

Machine a

For Machine a, the answer is called the **absolute length**, and it is equal to n **minus the number of cycles**.

One optimal algorithm (called **straigh selection sort** by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2431756$, we have

$$2431\mathbf{7}56 \overset{(57)}{\rightarrow} 2431\mathbf{6}57 \overset{(56)}{\rightarrow} 2\mathbf{4}31567 \overset{(24)}{\rightarrow} \mathbf{2}134567 \overset{(12)}{\rightarrow} 1234567$$

We obtain $w = 2431756 = (12)(24)(56)(57) = (124)(3)(576)$, and so $a(w) = 4 = n - \#\text{cycles}$.

Cycles

Machine a

For Machine a, the answer is called the **absolute length**, and it is equal to n **minus the number of cycles**.

One optimal algorithm (called **straigh selection sort** by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2431756$, we have

$$243\mathbf{1}\mathbf{7}56 \overset{(57)}{\to} 2431\mathbf{6}57 \overset{(56)}{\to} 2\mathbf{4}31567 \overset{(24)}{\to} \mathbf{2}134567 \overset{(12)}{\to} 1234567$$

We obtain $w = 2431756 = (12)(24)(56)(57) = (124)(3)(576)$, and so $a(w) = 4 = n - \#\text{cycles}$.

Straigh selection sort does not work for machine d

In the previous example $w = 2431756 = (12)(24)(56)(57)$ so

$$(7-5)+(6-5)+(4-2)+(2-1)=6$$

is the cost by using machine d.

- straight selection sort **optimizes** the number of transpositions needed to sort a permutation
- **but it does not** necessarily optimize the cost.

In fact for $2431756 = (12)(24)(67)(56)$ we have

$$(6-5)+(7-6)+(4-2)+(2-1)=5<6.$$

We can show that 5 is the lowest possible cost for such a w.

Sum of the sizes of exceedances

Machine d

For Machine d, the answer is called the **depth**. In 2015, Petersen–Tenner showed that it is equal to the **sum of the sizes of exceedances**, i.e.

$$d(w) = \sum_{w(i)>i} (w(i) - i).$$

They defined an optimal algorithm that given $w \in S_n$:

- finds an expression $w = t_1 \cdots t_r$ that realizes the depth of w;
- produces an expression with $r = a(w)$ transpositions.

Petersen –Tenner algorithm in S_n

$$w = 2431\mathbf{7}56$$

$$\begin{array}{ccc}
(67)\cdot & \downarrow & \cdot(57) \\
 & 2431\mathbf{6}57 & \\
(56)\cdot & \downarrow & \cdot(56) \\
 & 24\mathbf{3}1567 & \\
(14)\cdot & \downarrow & \cdot(24) \\
 & \mathbf{2}134567 & \\
(12)\cdot & \downarrow & \cdot(12) \\
 & e = 1234567 &
\end{array}$$

The associated expression for $w = 2431756$ is

$$w = (67)(12)(24)(56)$$

hence $d(w) = 5$, which coincides with

$$d(w) = \sum_{w(i)>i} (w(i) - i)$$

$$= (2-1) + (4-2) + (7-5) = 5.$$

Petersen –Tenner algorithm in S_n

$$\begin{array}{ccc} & w = 2431\mathbf{7}56 & \\ (67)\cdot & \downarrow & \cdot(57) \\ & 2431\mathbf{6}57 & \\ (56)\cdot & \downarrow & \cdot(56) \\ & 2\mathbf{4}31567 & \\ (14)\cdot & \downarrow & \cdot(24) \\ & \mathbf{2}134567 & \\ (12)\cdot & \downarrow & \cdot(12) \\ & e = 1234567 & \end{array}$$

The associated expression for $w = 2431756$ is

$$w = (67)(12)(24)(56)$$

hence $d(w) = 5$, which coincides with

$$d(w) = \sum_{w(i)>i} (w(i) - i)$$

$$= (2-1) + (4-2) + (7-5) = 5.$$

Petersen –Tenner algorithm in S_n

$$\begin{array}{ccc} & w = 2431\mathbf{7}56 & \\ (67)\cdot & \downarrow & \cdot(57) \\ & 2431\mathbf{6}57 & \\ (56)\cdot & \downarrow & \cdot(56) \\ & 2\mathbf{4}31567 & \\ (14)\cdot & \downarrow & \cdot(24) \\ & \mathbf{2}134567 & \\ (12)\cdot & \downarrow & \cdot(12) \\ & e = 1234567 & \end{array}$$

The associated expression for $w = 2431756$ is

$$w = (67)(12)(24)(56)$$

hence $d(w) = 5$, which coincides with

$$d(w) = \sum_{w(i)>i} (w(i) - i)$$

$$= (2-1) + (4-2) + (7-5) = 5.$$

Petersen –Tenner algorithm in S_n

$$\begin{array}{ccc}
 & w = 2431\mathbf{7}56 & \\
(67)\cdot & \downarrow & \cdot(57) \\
 & 2431\mathbf{6}57 & \\
(56)\cdot & \downarrow & \cdot(56) \\
 & 2\mathbf{4}31567 & \\
(14)\cdot & \downarrow & \cdot(24) \\
 & \mathbf{2}134567 & \\
(12)\cdot & \downarrow & \cdot(12) \\
 & e = 1234567 &
\end{array}$$

The associated expression for $w = 2431756$ is

$$w = (67)(12)(24)(56)$$

hence $d(w) = 5$, which coincides with

$$d(w) = \sum_{w(i)>i} (w(i) - i)$$

$$= (2-1) + (4-2) + (7-5) = 5.$$

Petersen –Tenner algorithm in S_n

$$\begin{array}{ccc}
 & w = 2431\mathbf{7}56 & \\
(67)\cdot & \downarrow & \cdot(57) \\
 & 2431\mathbf{6}57 & \\
(56)\cdot & \downarrow & \cdot(56) \\
 & 2\mathbf{4}31567 & \\
(14)\cdot & \downarrow & \cdot(24) \\
 & \mathbf{2}134567 & \\
(12)\cdot & \downarrow & \cdot(12) \\
 & e = 1234567 &
\end{array}$$

The associated expression for $w = 2431756$ is

$$w = (67)(12)(24)(56)$$

hence $d(w) = 5$, which coincides with

$$d(w) = \sum_{w(i)>i} (w(i) - i)$$

$$= (2 - 1) + (4 - 2) + (7 - 5) = 5.$$

Petersen –Tenner algorithm in S_n

$$\begin{array}{rcl}
 & w = 2431\mathbf{7}56 & \\
(67)\cdot & \downarrow & \cdot(57) \\
 & 2431\mathbf{6}57 & \\
(56)\cdot & \downarrow & \cdot(56) \\
 & 2\mathbf{4}31567 & \\
(14)\cdot & \downarrow & \cdot(24) \\
 & \mathbf{2}134567 & \\
(12)\cdot & \downarrow & \cdot(12) \\
 & e = 1234567 &
\end{array}$$

The associated expression for $w = 2431756$ is

$$w = (67)(12)(24)(56)$$

hence $d(w) = 5$, which coincides with

$$d(w) = \sum_{w(i)>i} (w(i) - i)$$

$$= (2-1) + (4-2) + (7-5) = 5.$$

From machines to Coxeter groups theory

Let (W, S) be a Coxeter system. Namely :

- S is a set of generators of the **Coxeter group** W
- the elements of S are involutions
- all the relations are of the form $(st)^{m_{st}} = 1$, where $m_{st} \in \mathbb{N}$

These relations can be rewritten as $s^2 = 1$ for all $s \in S$, and

$$\underbrace{sts\cdots}_{m_{st}} = \underbrace{tst\cdots}_{m_{st}},$$

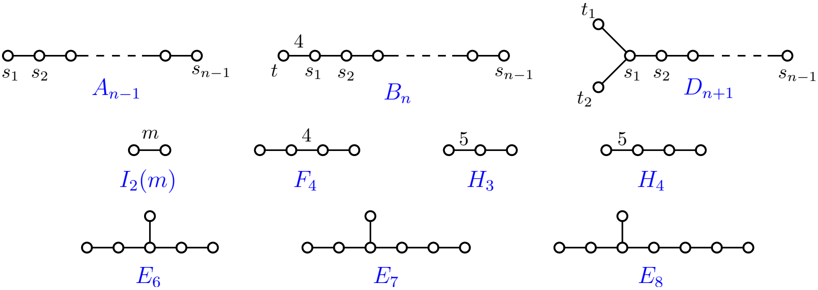
where $m_{st} < \infty$, the latter being called *braid relations*.

When $m_{st} = 2$, they are simply *commutation relations* $st = ts$.

Finite irreducible Coxeter systems

Theorem (Coxeter, 1934)

List of all the finite irreducible Coxeter groups.



Classification by Dynkin diagrams

Reduced expressions

Any element $w \in W$ can be written as $w = s_1 \cdots s_k$, with $s_i \in S$.

Length

$$\ell(w) = \min\{k \in \mathbb{N} \mid w = s_1 \cdots s_k \text{ for } s_i \in S\}$$

An expression $w = s_1 \cdots s_k$ of minimal length is called **reduced**.

Set of reflections of W

$$T = \{wsw^{-1} \mid s \in S, w \in W\}.$$

Absolute length

$$a(w) = \min\{k \in \mathbb{N} \mid w = t_1 \cdots t_k \text{ for } t_i \in T\}$$

Reduced expressions

Any element $w \in W$ can be written as $w = s_1 \cdots s_k$, with $s_i \in S$.

Length

$$\ell(w) = \min\{k \in \mathbb{N} \mid w = s_1 \cdots s_k \text{ for } s_i \in S\}$$

An expression $w = s_1 \cdots s_k$ of minimal length is called **reduced**.

Set of reflections of W

$$T = \{wsw^{-1} \mid s \in S, w \in W\}.$$

Absolute length

$$a(w) = \min\{k \in \mathbb{N} \mid w = t_1 \cdots t_k \text{ for } t_i \in T\}$$

Bruhat graph and weak order

Bruhat graph

The **Bruhat graph** is the directed graph whose nodes are the elements of W and whose edges are given by :
$u \xrightarrow{t} w$ means that $w = ut$ for some $t \in T$, and $\ell(u) < \ell(w)$.

Bruhat order

The **Bruhat order** is the transitive closure of the primitive relations $u \xrightarrow{t} ut$, where $t \in T$.

Weak order

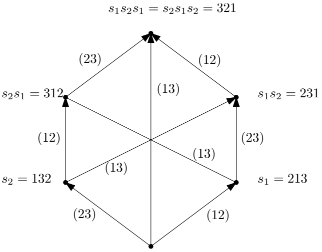
The **(right) weak order** is the transitive closure of the primitive relations $u \xrightarrow{s} us$, where $s \in S$.

Symmetric group

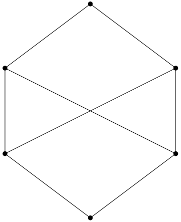
The symmetric group is a Coxeter group of type A_{n-1}

- $S = \{s_1, \ldots, s_{n-1}\}$ with $s_i = (i, i+1)$ is the **simple transposition** exchanging i and $i+1$.
  - $s_i^2 = Id$
  - $s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}$
  - $s_i s_j = s_j s_i$ if $|i-j| \geq 2$
- The **set of reflections** is $T = \{t_{ij} = (i,j) \mid 1 \leq i < j \leq n\}$, where (i,j) is the transposition exchanging i and j.

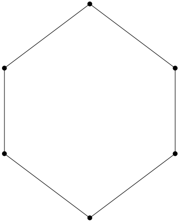
Symmetric group : Bruhat graph



Bruhat graph of S_3



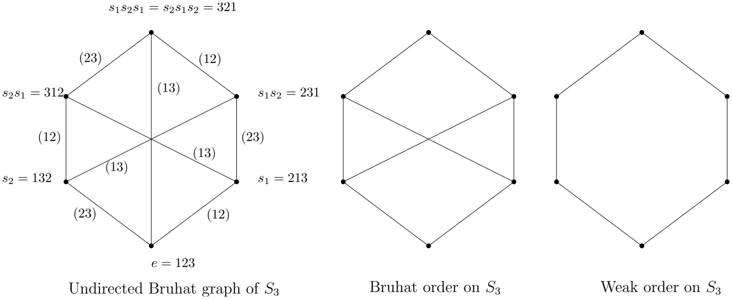
Bruhat order on S_3



Weak order on S_3

- The length ℓ is the rank function of such posets.
- The edges are labeled by reflections.

Symmetric group : Bruhat graph



Undirected Bruhat graph of S_3 Bruhat order on S_3 Weak order on S_3

- The length ℓ is the rank function of such posets.
- The edges are labeled by reflections.

Algebraic motivation : depth in terms of roots

Let $\Phi = \Phi^+ \cup \Phi^-$ be the **root system** for (W, S).

The depth $dp(\beta)$ of a positive root is a well-known parameter.

Since there is

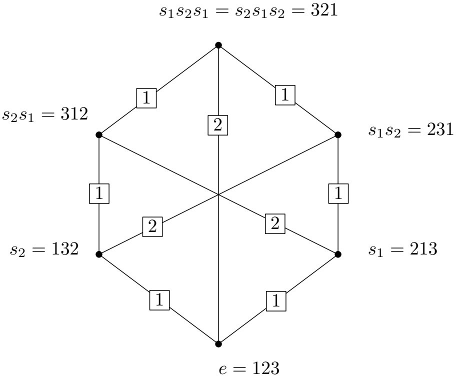
- a bijection between Φ^+ and T : $\beta \mapsto t_\beta$
- by letting $d(t_\beta) = dp(\beta) = \frac{1+\ell(t_\beta)}{2}$ **(costs of the machine d)**

Petersen and Tenner defined

Definition of depth

$$d(w) = \min \left\{ \sum_{i=1}^{k} \frac{1+\ell(t_i)}{2} \mid w = t_1 \cdots t_k \text{ for } t_i \in T \right\}.$$

Undirected paths in the weighted Bruhat graph

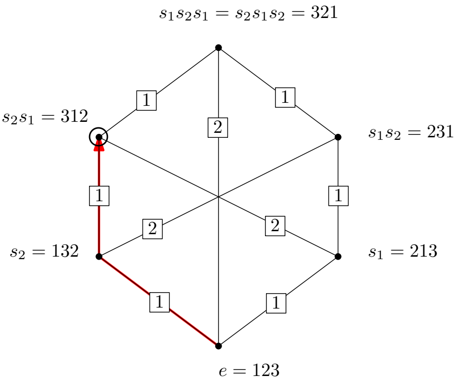


Undirected Bruhat graph of S_3

This means that the depth of w is equal to the minimal cost of an undirected path going from e to w in the Bruhat graph of W where each edge is labeled by

$$t \longrightarrow (1 + \ell(t))/2$$

Undirected paths in the weighted Bruhat order

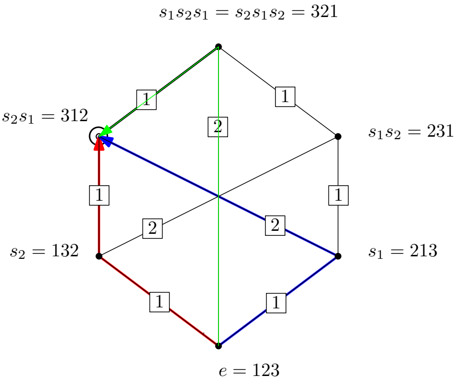


Undirected Bruhat graph of S_3

This means that the depth of w is equal to the minimal cost of an undirected path going from e to w in the Bruhat graph of W where each edge is labeled by

$$t \longrightarrow (1 + \ell(t))/2$$

Undirected paths in the weighted Bruhat order



Undirected Bruhat graph of S_3

This means that the depth of w is equal to the minimal cost of an undirected path going from e to w in the Bruhat graph of W where each edge is labeled by

$$t \longrightarrow (1 + \ell(t))/2$$

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w) + \ell(w)}{2} \leq d(w) \leq \ell(w).$$

- The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations.
- The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations.
- It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w)+\ell(w)}{2} \leq d(w) \leq \ell(w).$$

- The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations.
- The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations.
- It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w) + \ell(w)}{2} \leq d(w) \leq \ell(w).$$

- The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations.
- The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations.
- It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w) + \ell(w)}{2} \leq d(w) \leq \ell(w).$$

- The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations.
- The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations.
- It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Petersen-Tenner questions

Let (W, S) be a Coxeter group, and $B(W)$ its associated directed Bruhat graph.

Question

Is it true that

- $d(w) = min\left\{ \sum_{i=1}^{k} d_i \mid \exists\ e \xrightarrow{d_1} \cdots \xrightarrow{d_k} w \text{ in } B(W) \right\}$ *?*
- *And if so, can be the path chosen so that it has* $a(w)$ *edges ?*

Question

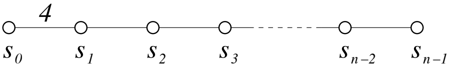
Are there explicit formulas for depth for Coxeter groups of types B and D ?

The group of signed permutations B_n

A **signed permutation** is a permutation w on the set $\{\pm 1, \ldots, \pm n\}$ with the property that $w(-i) = -w(i)$ for all i.

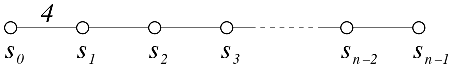
To define a signed permutations it suffices to specify $w(i)$ for $i > 0$. For example $[\bar{3}, 5, 1, \bar{4}, \bar{2}] \in B_5$.

The Dynkin diagram of type B_n is :



The group B_n has a combinatorial interpretation in terms of signed permutations.

Coxeter group of type B



Set

$$\begin{aligned} s_0 &:= [\bar{1}, 2, \ldots, n] = (-1, 1), \\ s_i &:= [1, \ldots, i-1, i+1, i, i+2, \ldots, n] = (i, i+1)(-i, -i-1), \\ \text{and} \quad & S_B := \{s_0, s_1, \ldots, s_{n-1}\} \end{aligned}$$

Then (B_n, S_B) forms a Coxeter system of type B.
For example

$$B_2 = \{[1,2], [\bar{1},2], [1,\bar{2}], [\bar{1},\bar{2}], [2,1], [\bar{2},1], [2,\bar{1}], [\bar{2},\bar{1}]\}.$$

Depth machine for B_n

The set of reflections of B_n is given by

$$T = \{t_{ij}, t_{\bar{i}j} \mid 1 \leq i < j \leq n\} \cup \{t_{\bar{i}i} \mid i \in [n]\},$$

where $t_{ij} = (i, j)(\bar{i}, \bar{j})$, $t_{\bar{i}j} = (\bar{i}, j)(i, \bar{j})$ and $t_{\bar{i}i} = (\bar{i}, i)$.

Machine d can :

- Shuffling (t_{ij}) : swap a pair of entries at positions i and j, with cost $j - i$ (as for the symmetric group)
- Double unsigning $(t_{\bar{i}j})$: swap a pair of entries at positions i and j and change both signs, with cost $i + j - 1$
- Single unsigning $(t_{\bar{i}i})$: change the sign of the entry at position i, with cost i.

Oddness of a signed permutation

First decompose the signed permutation as sum ($\oplus$) of **indecomposable** signed permutations (by ignoring the signs).

For example $2\bar{4}3\bar{6}7\bar{5}1$ is indecomposable while $2\bar{4}3\bar{1}7\bar{5}6 = 2\bar{4}3\bar{1} \oplus 3\bar{1}2$ is the sum decomposition.

Definition

Given a signed permutation w, define the **oddness** of w to be the number of blocks in the sum decomposition with an odd number of signed elements, denoted $o(w)$.

The negative identity $\bar{1} \cdots \bar{n}$ is the oddest element, with oddness n.

In the previous example $o(2\bar{4}3\bar{1}7\bar{5}6) = 2\bar{4}3\bar{1} \oplus 3\bar{1}2 = 1$.

Oddness of a signed permutation

First decompose the signed permutation as sum ($\oplus$) of **indecomposable** signed permutations (by ignoring the signs).

For example $2\bar{4}3\bar{6}7\bar{5}1$ is indecomposable while $2\bar{4}3\bar{1}7\bar{5}6 = 2\bar{4}3\bar{1} \oplus 3\bar{1}2$ is the sum decomposition.

Definition

Given a signed permutation w, define the **oddness** of w to be the number of blocks in the sum decomposition with an odd number of signed elements, denoted $o(w)$.

The negative identity $\bar{1} \cdots \bar{n}$ is the oddest element, with oddness n.

In the previous example $o(2\bar{4}3\bar{1}7\bar{5}6) = 2\bar{4}3\bar{1} \oplus 3\bar{1}2 = 1$.

Depth for a signed permutation

Theorem (BBNW, 2016)

Let $w \in B_n$. Then

$$d(w) = \sum_{w(i)>i} (w(i) - i) + \sum_{w(i)<0} |w(i)| + \frac{o(w) - neg(w)}{2}.$$

Single unsigning moves are slightly expensive, and $o(w)$ counts how many times they need to be used.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at **least two negative entries**, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is **one negative entry**, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at **least two negative entries**, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is **one negative entry**, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at **least two negative entries**, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is **one negative entry**, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at **least two negative entries**, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is **one negative entry**, apply a single unsigning move the negative entry, and go back to Step 1.

Restriction to S_n : application of Step 1.

	$Petersen - Tenner$		$New\ algorithm$	
	$w = 2431\mathbf{7}56$		$w = 2431\mathbf{7}56$	
$(67)\cdot$	$\downarrow$		$\downarrow$	$\cdot(56)$
	$2431\mathbf{6}57$		$24315\mathbf{7}6$	
	$\downarrow$	$\cdot(56)$	$\downarrow$	$\cdot(67)$
	$2\mathbf{4}31567$		$2\mathbf{4}31567$	
	$\downarrow$	$\cdot(24)$	$\downarrow$	$\cdot(24)$
	$\mathbf{2}134567$		$\mathbf{2}134567$	
$(12)\cdot$	$\downarrow$		$\downarrow$	$\cdot(12)$
	$e = 1234567$		$e = 1234567$	

Restriction to S_n : application of Step 1.

$$\begin{array}{cccccc}
 & Petersen-Tenner & & & New\ \ algorithm & \\
 & w=2431\mathbf{7}56 & & & w=2431\mathbf{7}56 & \\
(67)\cdot & \downarrow & & & \downarrow & \cdot(56) \\
 & 2431\mathbf{6}57 & & & 24315\mathbf{7}6 & \\
 & \downarrow & \cdot(56) & & \downarrow & \cdot(67) \\
 & 2\mathbf{4}31567 & & & 2\mathbf{4}31567 & \\
 & \downarrow & \cdot(24) & & \downarrow & \cdot(24) \\
 & \mathbf{2}134567 & & & \mathbf{2}134567 & \\
(12)\cdot & \downarrow & & & \downarrow & \cdot(12) \\
 & e=1234567 & & & e=1234567 &
\end{array}$$

Example of the algorithm : Steps 1,2,3

For $w = 2\bar{4}3\bar{1}7\bar{5}6 = [2\bar{4}3\bar{1}] \oplus [3\bar{1}2]$, the formula

$$d(w) = \sum_{w(i)>i} (w(i) - i) + \sum_{w(i)<0} |w(i)| + \frac{o(w) - neg(w)}{2}$$

gives $d(w) = (1+2) + (4+1+5) + (1-3)/2 = 12$

We apply the algorithm separately to the two blocks:

$$2\bar{4}3\bar{1}\_\mathbf{7\bar{5}}6 \xrightarrow[1]{t_{45}} 2\bar{4}3\bar{1}\_\bar{5}\mathbf{76} \xrightarrow[1]{t_{67}} 2\bar{4}3\bar{1}\_\mathbf{\bar{5}}67 \xrightarrow[5]{t_{\bar{5}5}} \mathbf{2\bar{4}}3\bar{1}\_567$$

$$\xrightarrow[1]{t_{12}} \mathbf{\bar{4}}23\mathbf{\bar{1}}\_567 \xrightarrow[4]{t_{\bar{1}4}} 1234567$$

Increasing paths, reduced factorizations, and weak order

Our algorithms provide factorizations

$$w = t_1 \cdots t_k \text{ such that } d(w) = d(t_1) + \cdots + d(t_k)$$

with the properties that:

- $e \xrightarrow{d_1} t_1 \xrightarrow{d_2} t_1 t_2 \xrightarrow{d_3} \ldots \xrightarrow{d_k} w$ is an **increasing path** in the directed Bruhat graph;
- $k \neq a(w)$ (different from Petersen–Tenner);
- $\ell(w) = \ell(t_1) + \cdots + \ell(t_k)$. When this happens we say that the depth is **realized by a reduced factorization**;
- Moreover $e <_R t_1 <_R t_1 t_2 <_R \ldots <_R t_1 t_2 \cdots t_k$, where $<_R$ denotes the **right weak order**.

Increasing paths, reduced factorizations, and weak order

Our algorithms provide factorizations

$$w = t_1 \cdots t_k \text{ such that } d(w) = d(t_1) + \cdots + d(t_k)$$

with the properties that:

- $e \xrightarrow{d_1} t_1 \xrightarrow{d_2} t_1 t_2 \xrightarrow{d_3} \ldots \xrightarrow{d_k} w$ is an **increasing path** in the directed Bruhat graph;
- $k \neq a(w)$ (different from Petersen–Tenner);
- $\ell(w) = \ell(t_1) + \cdots + \ell(t_k)$. When this happens we say that the depth is **realized by a reduced factorization**;
- Moreover $e <_R t_1 <_R t_1 t_2 <_R \ldots <_R t_1 t_2 \cdots t_k$, where $<_R$ denotes the **right weak order**.

Increasing paths, reduced factorizations, and weak order

Our algorithms provide factorizations

$$w = t_1 \cdots t_k \text{ such that } d(w) = d(t_1) + \cdots + d(t_k)$$

with the properties that:

- $e \xrightarrow{d_1} t_1 \xrightarrow{d_2} t_1 t_2 \xrightarrow{d_3} \ldots \xrightarrow{d_k} w$ is an **increasing path** in the directed Bruhat graph;
- $k \neq a(w)$ (different from Petersen–Tenner);
- $\ell(w) = \ell(t_1) + \cdots + \ell(t_k)$. When this happens we say that the depth is **realized by a reduced factorization**;
- Moreover $e <_R t_1 <_R t_1 t_2 <_R \ldots <_R t_1 t_2 \cdots t_k$, where $<_R$ denotes the **right weak order**.

Increasing paths, reduced factorizations, and weak order

Our algorithms provide factorizations

$$w = t_1 \cdots t_k \text{ such that } d(w) = d(t_1) + \cdots + d(t_k)$$

with the properties that:

- $e \xrightarrow{d_1} t_1 \xrightarrow{d_2} t_1 t_2 \xrightarrow{d_3} \ldots \xrightarrow{d_k} w$ is an **increasing path** in the directed Bruhat graph;
- $k \neq a(w)$ (different from Petersen–Tenner);
- $\ell(w) = \ell(t_1) + \cdots + \ell(t_k)$. When this happens we say that the depth is **realized by a reduced factorization**;
- Moreover $e <_R t_1 <_R t_1 t_2 <_R \ldots <_R t_1 t_2 \cdots t_k$, where $<_R$ denotes the **right weak order**.

Increasing paths, reduced factorizations, and weak order

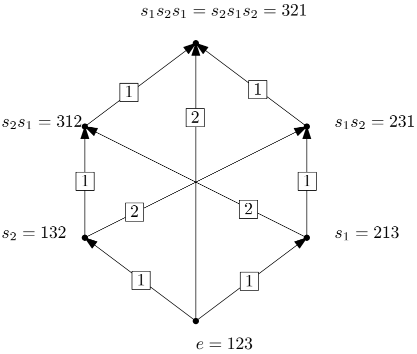
Our algorithms provide factorizations

$$w = t_1 \cdots t_k \text{ such that } d(w) = d(t_1) + \cdots + d(t_k)$$

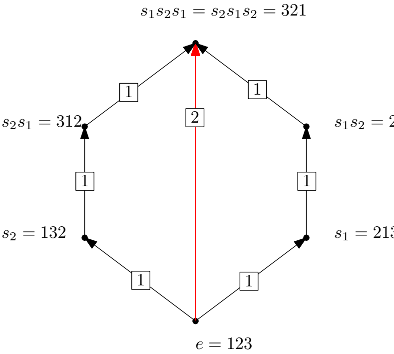
with the properties that:

- $e \xrightarrow{d_1} t_1 \xrightarrow{d_2} t_1 t_2 \xrightarrow{d_3} \ldots \xrightarrow{d_k} w$ is an **increasing path** in the directed Bruhat graph;
- $k \neq a(w)$ (different from Petersen–Tenner);
- $\ell(w) = \ell(t_1) + \cdots + \ell(t_k)$. When this happens we say that the depth is **realized by a reduced factorization**;
- Moreover $e <_R t_1 <_R t_1 t_2 <_R \ldots <_R t_1 t_2 \cdots t_k$, where $<_R$ denotes the **right weak order**.

Directed paths in the weighted Bruhat order



Directed Bruhat graph of S_3



Directed Weak graph of S_3

Comparing length and depth

An element in a Coxeter group (W, S) is **short-braid-avoiding** if no reduced decomposition (product of simple reflections realizing w) has a consecutive subexpression $s_i s_j s_i$, with $s_i, s_j \in S$.

Theorem (BBNW, 2016)

Let (W, S) any Coxeter system. Then $d(w) = \ell(w)$ if and only if the depth of w is realized by a reduced factorization and w is short-braid-avoiding.

Since the depth is always realized by a reduced factorization in S_n and B_n, this shows that $d(w) = \ell(w)$ in those groups if and only if w is short-braid-avoiding.

Comparing length and depth

An element in a Coxeter group (W, S) is **short-braid-avoiding** if no reduced decomposition (product of simple reflections realizing w) has a consecutive subexpression $s_i s_j s_i$, with $s_i, s_j \in S$.

Theorem (BBNW, 2016)

Let (W, S) any Coxeter system. Then $d(w) = \ell(w)$ if and only if the depth of w is realized by a reduced factorization and w is short-braid-avoiding.

Since the depth is always realized by a reduced factorization in S_n and B_n, this shows that $d(w) = \ell(w)$ in those groups if and only if w is short-braid-avoiding.

Comparing length and depth

An element in a Coxeter group (W, S) is **short-braid-avoiding** if no reduced decomposition (product of simple reflections realizing w) has a consecutive subexpression $s_i s_j s_i$, with $s_i, s_j \in S$.

Theorem (BBNW, 2016)

Let (W, S) any Coxeter system. Then $d(w) = \ell(w)$ if and only if the depth of w is realized by a reduced factorization and w is short-braid-avoiding.

Since the depth is always realized by a reduced factorization in S_n and B_n, this shows that $d(w) = \ell(w)$ in those groups if and only if w is short-braid-avoiding.

Short-braid-avoidance in S_n, B_n, and D_n

In 1995, Billey-Jockusch-Stanley showed that a permutation is **321-avoiding if and only if is short-braid avoiding**. Such permutations are usually called **fully commutative**.

For permutations, this reproves the Petersen–Tenner theorem that $d(w) = \ell(w)$ if and only if w is 321-avoiding.

In B_n, short-braid-avoiding is equivalent to Stembridge's notion of **fully commutative top-and-bottom**, which is characterized by avoiding $1\bar{2}$, $\bar{1}\bar{2}$, $\bar{2}\bar{1}$, $\bar{3}21$, $\bar{3}2\bar{1}$, and 321.

Short-braid-avoidance in S_n, B_n, and D_n

In 1995, Billey-Jockusch-Stanley showed that a permutation is **321-avoiding if and only if is short-braid avoiding**. Such permutations are usually called **fully commutative**.

For permutations, this reproves the Petersen–Tenner theorem that $d(w) = \ell(w)$ if and only if w is 321-avoiding.

In B_n, short-braid-avoiding is equivalent to Stembridge's notion of **fully commutative top-and-bottom**, which is characterized by avoiding $1\bar{2}$, $\bar{1}\bar{2}$, $\bar{2}\bar{1}$, $\bar{3}21$, $\bar{3}2\bar{1}$, and 321.

Achieving the lower bound

Definition

An element $w \in W$ is **boolean**, if some (and hence any) reduced decomposition of w has no repeated simple reflections.

Theorem

We have that $a(w) = d(w)$ (and hence both are equal to $\ell(w)$) if and only of w is boolean. These elements are characterized by avoiding 10 patterns for B_n (Tenner).

The more general question of when $d(w) = (a(w) + \ell(w))/2$ seems hard and is not characterized by pattern avoidance.

Problems

- How many elements of B_n and D_n have depth k ?
- Characterize depth for elements in affine Coxeter groups.
- Is depth universally realized by reduced factorizations for all Coxeter groups? If so, is there a uniform proof ? If not, can one characterize the elements whose depth is realized by a reduced factorization ?
- Is depth the rank function of an interesting poset ?

The End

Thank you for your attention!

The group D_n

$$D_n = \{w \in B_n \mid neg(w) \equiv 0 \text{ (mod) } 2\}.$$

The set of reflections of D_n is given by

$$T = \{t_{ij}, t_{\bar{i}j} \mid 1 \leq i < j \leq n\}.$$

Machine d can :

- Shuffling (t_{ij}) : swap a pair of entries at positions i and j, with cost $j - i$ (as for the symmetric group)
- Double unsigning $(t_{\bar{i}j})$: swap a pair of entries at positions i and j and change both signs, with cost $i + j - 2$ (1 less then type B)
- Single unsigning $(t_{\bar{i}i})$: are banned !

Sum decompositions for D_n

For D_n, we need to distinguish between two types of sum decompositions. A **type D decomposition** requires that each block has an even number of negative entries, while a **type B decomposition** does not.

If $w = \bar{2}134\overline{578}6$, then
$w = \bar{2}134\bar{5} \oplus \overline{23}1$ is the **type D decomposition**,
$w = \bar{2}1 \oplus 1 \oplus 1 \oplus \bar{1} \oplus \overline{23}1$ is the **type B decomposition**.

Definition

Define **oddness** in type D (denoted $o^D(w)$) as the number of type B blocks minus the number of type D blocks (so $o^D(w) = 3$).

Sum decompositions for D_n

For D_n, we need to distinguish between two types of sum decompositions. A **type D decomposition** requires that each block has an even number of negative entries, while a **type B decomposition** does not.

If $w = \overline{2}134\overline{578}6$, then
$w = \overline{2}134\overline{5} \oplus \overline{23}1$ is the **type D decomposition**,
$w = \overline{2}1 \oplus 1 \oplus 1 \oplus \overline{1} \oplus \overline{23}1$ is the **type B decomposition**.

Definition

Define **oddness** in type D (denoted $o^D(w)$) as the number of type B blocks minus the number of type D blocks (so $o^D(w) = 3$).

Sum decompositions for D_n

For D_n, we need to distinguish between two types of sum decompositions. A **type D decomposition** requires that each block has an even number of negative entries, while a **type B decomposition** does not.

If $w = \overline{2}1345\overline{578}6$, then
$w = \overline{2}134\overline{5} \oplus \overline{23}1$ is the **type D decomposition**,
$w = \overline{2}1 \oplus 1 \oplus 1 \oplus \overline{1} \oplus \overline{23}1$ is the **type B decomposition**.

Definition

Define **oddness** in type D (denoted $o^D(w)$) as the number of type B blocks minus the number of type D blocks (so $o^D(w) = 3$).

Depth for an even signed permutation

Theorem (BBNW, 2015)

Let $w \in D_n$. *Then*

$$d(w) = \left(\sum_{w(i)>i} (w(i) - i) \right) + \left(\sum_{w(i)<0} |w(i)| \right) + \left(o^D(w) - neg(w) \right)$$

The D-oddness counts the “wasted” moves that are needed to join type B blocks so that we can perform the needed double unsigning moves.

Example in type D

$$[3,\bar{1},2,6,\bar{5},4] \xrightarrow{t_{45}} [3,\bar{1},2,\bar{5},6,4] \xrightarrow{t_{56}} [3,\bar{1},2,\bar{5},4,6]$$

Unite two B-blocks :

$$\xrightarrow{t_{34}} [3,\bar{1},\bar{5},2,4,6]$$

Shuffle inside the united block (which is now B-indecomposable):

$$\xrightarrow{t_{13}} [\bar{5},\bar{1},3,2,4,6]$$

Then double unsign:

$$\xrightarrow{t_{\bar{1}2}} [1,5,3,2,4,6]$$

and shuffle toward the end:

$$\xrightarrow{t_{24}} [1,2,3,5,4,6] \xrightarrow{t_{45}} [1,2,3,4,5,6]$$

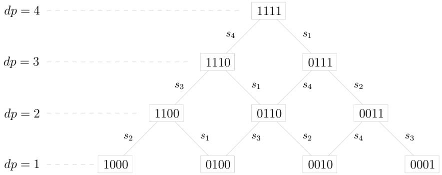
Depth of a positive root

Let $\Phi = \Phi^+ \cup \Phi^-$ be the **root system** for (W, S) and Δ the simple roots.

The **depth** $dp(\beta)$ of a positive root $\beta \in \Phi^+$ is defined as

$$dp(\beta) = \min\{k \mid s_1 \cdots s_k(\beta) \in \Phi^-, s_j \in S\}.$$

The depth is the rank function of the **root poset**. For A_{n-1}, $\Phi^+ = \{e_j - e_i \mid 1 \leq i < j \leq n\}$, $\Delta = \{\alpha_i = e_{i+1} - e_i \mid i \in [n-1]\}$.



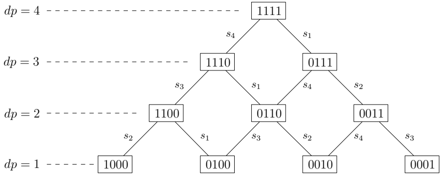
Root poset of A_4

Depth of a positive root

Let $\Phi = \Phi^+ \cup \Phi^-$ be the **root system** for (W, S) and Δ the simple roots.
The **depth** $dp(\beta)$ of a positive root $\beta \in \Phi^+$ is defined as

$$dp(\beta) = \min\{k \mid s_1 \cdots s_k(\beta) \in \Phi^-, s_j \in S\}.$$

The depth is the rank function of the **root poset**. For A_{n-1},
$\Phi^+ = \{e_j - e_i \mid 1 \leq i < j \leq n\}$, $\Delta = \{\alpha_i = e_{i+1} - e_i \mid i \in [n-1]\}$.



Root poset of A_4

Depth in terms of roots

There is a bijection between positive roots and reflections,

$$\Phi^+ \longleftrightarrow T$$

and denote by t_β the reflection corresponding to the root β.

Definition (Depth of $w \in W$)

For any $w \in W$ Petersen and Tenner defined

$$d(w) = \min\left\{\sum_{i=1}^{k} dp(\beta_i) \mid w = t_{\beta_1} \cdots t_{\beta_k},\ t_{\beta_i} \in T\right\}.$$

Depth in terms of roots

There is a bijection between positive roots and reflections,

$$\Phi^+ \longleftrightarrow T$$

and denote by t_β the reflection corresponding to the root β.

Definition (Depth of $w \in W$)

For any $w \in W$ Petersen and Tenner defined

$$d(w) = \min \left\{ \sum_{i=1}^{k} dp(\beta_i) \mid w = t_{\beta_1} \cdots t_{\beta_k},\ t_{\beta_i} \in T \right\}.$$

Algebraic meaning and algebraic motivation

Since for any reflection one has

$$d(t_\beta) = dp(\beta) = \frac{1+\ell(t_\beta)}{2},$$

then

$$d(w) = \min\left\{\sum_{i=1}^{k} \frac{1+\ell(t_i)}{2} \mid w = t_1 \cdots t_k \text{ for } t_i \in T\right\}.$$

Algebraic meaning and algebraic motivation

Since for any reflection one has

$$d(t_\beta) = dp(\beta) = \frac{1+\ell(t_\beta)}{2},$$

then

$$d(w) = \min\left\{\sum_{i=1}^{k} \frac{1+\ell(t_i)}{2} \mid w = t_1 \cdots t_k \text{ for } t_i \in T\right\}.$$

Algebraic meaning and algebraic motivation

Since for any reflection one has

$$d(t_\beta) = dp(\beta) = \frac{1 + \ell(t_\beta)}{2},$$

then

$$d(w) = \min \left\{ \sum_{i=1}^{k} \frac{1 + \ell(t_i)}{2} \mid w = t_1 \cdots t_k \text{ for } t_i \in T \right\}.$$