O(1) TIME GENERATION OF MULTISET PERMUTATION BY PREFIX REVERSALS

MULTISET: A SET IN WHICH ELEMENTS MAY BE REPEATED.

e.g. { 1,1,2,2,4,3,3,3}

If E is a Multiset permutation can be described by { $M_{E_1}, M_{E_2}, \dots \dots . M_{E_{LAST}}$ }

>(S): Let S, be a string $(S_1, S_2, \dots \dots . . S_n)$. Then >(S) will give an integer for non-increasing length.

Example:

Rough 1<3
⇨ (4221 . 3910)

>(4221 3910)

= 4

$\pi_j\ (S)$: prefix reversal 2 times in order.

1 ⟶ j
2 ⟶ j

[see Later, By BLL $\pi_j\ (S)$ can be computed in O(1) Time.
∵ O(1) = O(1) + O(1)]

e.g. π_4 (3 2 4 5 1 3 4 1)
1 2 3 4

= (5 3 2 4 1 3 4 1)

Rough
5 4 2 3 . 1341
1 2 3 4
5 3 2 4 . 1341

tail (E) = String arranged in non-increasing order.

let, function mapping $\Delta: M_{E_K} \rightarrow M_{E_{K+1}}$
K ={1,2, … …(LAET-1)}

means, previous string generates next string.

we've cool – lex definition

$$C_E = \bigoplus_{P=0}^{|M_E|-1} \Delta^P(\text{tail (E)}) \quad \text{——— (1)}$$

Δ: let S= $S_1, S_2, \dots \dots \dots . . S_n$; $i = >(S)$

$$\Delta_i(S) = \begin{cases} \pi_{i+1}(S) & if\ i \le n-2\ \&\&\ S_{i+2} > S_i \\ \pi_{i+2}(S) & if\ i \le n-2\ \&\&\ S_{i+2} \le S_i \\ \pi_n(S) & otherwise (i \ge n-1) \end{cases}$$

$\Delta^0(S) = S$

$\Delta^K(S) = \Delta(\Delta^{K-1}(S))$

We'll get this, relation simply replacing.

prefix shift by prefix reversals. As $\pi_j\ (S)$ and $\sigma_j\ (S)$ are function equivalent.

for example,

$\sigma_j\ (S)$= prefix shift

$\sigma_3(12345)$= (31245)

$\pi_j\ (S) = prefix\ reversal$

$\pi_3 = \pi_3(12345)$
= (31245)

Rough
1 2 3 4 5
⟵
3 2 1 4 5
⟶
3 1 2 4 5

Here in (1) ⊕ is used for appending list,

e.g. $\bigoplus_{i=1}^{3} L_i = L_1, L_2, L_3$
= 11, 22, 33
(3 strings).

Let,
$L_1 = 11,$
$L_2 = 22,$
$L_3 = 33$

Now at (1) we need index of π as π_j gives operating behavior of Δ. This index will be given from previous research work as follows,

Let, S=$S_1, S_2, \dots \dots \dots . . S_n = tail$ then j=n

let, $\Delta_{j=n}(S) = S'_1\ S'_2 \dots \dots . . S'_n$

Then

$$j_{next} \Rightarrow (\pi_{j=n}(S)) = \begin{cases} 1 & if\ S'_1 < S'_2 \\ >(S)+1 & if\ S'_1 \ge S'_2 \end{cases}$$

[initially j=n]

e.g E= 322 11 (tail)

Next => 1 3 221

Rough n=5
1 1 2 2 3
1 3 2 2 1

1<3

$S_1' < S_2'$

So, j=1

π_1 (1 3 2 2 1)

Next => = (3 1 2 2 1)

etc.

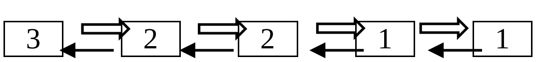
Boustrophedon linked list one kind of doubly linked list hiding information of forward and backward pointer, it can provide O(1) Time string reversals where traditional limit was ⊖ (n).

Using BLL $\pi_j\ (S)$ can be computed in O(1) Time.

So, our final Algorithm for looplessly generating multiset permutation by prefix reversal can be described as below, [in cool-lex order]

1) Take tail of E (Make a BLL String representation)
2) do{
3) compute j= > (BLL) [2 comparison at most]
4) compute Δ_j(BLL) [2 comparison at most]
5) Visit (BLL)
6) { while (Again tail not generated);

Cool – tex tested generation



⇒ White pointer

→ Black pointer

It follows circular minimal change order it. Also uses “constant number of additional variables.”

[consumer-producer sense]

32211	31122	12231
13221	23112	21231
31221	12312	22131
23121	21312	12213
12321	12132	21213
21321	11232	12123
32121	21132	11223
12312	32112	21123
31212	23211	22113
13122	22311	
11322		