# An Upper Bound for Sorting $R_n$ with LRE

Sai Satwik Kuppili\* and Bhadrachalam Chitturi<sup>†</sup>

\* Dept. of Computer Science and Engineering, Amrita University

Amritapuri, India

<sup>†</sup>Dept. of Computer Science, University of Texas at Dallas

Texas, USA

\*VMware Inc. Pune, India

Email: \*satwik.kuppili@gmail.com, <sup>†</sup>chalam@utdallas.edu

<sup>1</sup> Abstract—A permutation  $\pi$  over alphabet  $\Sigma = 1, 2, 3, \ldots, n$ , is a sequence where every element x in  $\Sigma$  occurs exactly once.  $S_n$  is the symmetric group consisting of all permutations of length n defined over  $\Sigma$ .  $I_n = (1, 2, 3, ..., n)$  and  $R_n =$  $(n, n-1, n-2, \ldots, 2, 1)$  are identity (i.e. sorted) and reverse permutations respectively. An operation, that we call as an LREoperation, has been defined in OEIS with identity A186752. This operation is constituted by three generators: left-rotation, rightrotation and transposition(1,2). We call transposition(1,2) that swaps the two leftmost elements as Exchange. The minimum number of moves required to transform  $R_n$  into  $I_n$  with LRE operation are known for n < 11 as listed in OEIS with sequence number A186752. For this problem no upper bound is known. OEIS sequence A186783 gives the conjectured diameter of the symmetric group  $S_n$  when generated by LRE operations [1]. The contributions of this article are: (a) The first non-trivial upper bound for the number of moves required to sort  $R_n$  with LRE; (b) a tighter upper bound for the number of moves required to sort  $R_n$  with LRE; and (c) the minimum number of moves required to sort  $R_{10}$  and  $R_{11}$  have been computed. Here we are computing an upper bound of the diameter of Cayley graph generated by LRE operation. Cayley graphs are employed in computer interconnection networks to model efficient parallel architectures. The diameter of the network corresponds to the maximum delay in the network.

*Index Terms*—Permutation, Sorting, Left Rotate, Right Rotate, Exchange, Symmetric Group, Upper Bound, Cayley Graphs.

#### I. INTRODUCTION

The following problem is from OEIS with sequence number A186752: "Length of minimum representation of the permutation  $[n, n-1, \ldots, 1]$  as the product of transpositions (1, 2)and left and right rotations (1, 2, ..., n). [1]." We call this operation as *LRE*. *LRE* operation consists of following three generators: (i) LeftRotate that cyclically shifts all elements to left by one position, (ii) RightRotate that cyclically shifts all elements to right by one position and (iii) Exchange that swaps the leftmost two elements of the permutation. The mentioned operations are abbreviated as L, R and Erespectively.  $R_n$  denotes  $(n, n-1, \dots, 2, 1)$  whereas  $I_n$  denotes the sorted order or identity permutation: (1, 2, ..., n). Sorting a permutation  $\pi$  in this article refers to transforming  $\pi$  into  $I_n$ with *LRE* operation. The alphabet is  $\Sigma = (1, 2, 3, ..., n)$ . [2], [3] studied a more restricted version of this problem, i.e. LE operation where the operation R is disallowed and appears in OEIS with sequence number A048200 [1]. We note that the results of [2], [3] are applicable to RE operation (that has not been studied) due to symmetry. We seek to obtain an upper bound on the length of generator sequence that transforms  $R_n$  with LRE into  $I_n$ .

The optimum number of moves to sort  $R_n$  with LRE are known only for  $n \leq 11$  (n = 10 and n = 11 are our contributions). We give the first non-trivial upper bound to sort  $R_n$  with LRE.

Let  $\pi[1 \cdots n]$  be the array containing the input permutation. The element at an index i is denoted by  $\pi[i]$ . Initially for all  $i, \pi[i] = R_n[i]$ . We define a permutation  $K_{r,n} \in S_n$ as follows. The elements  $n - (r - 1), n - (r - 2), \dots n$ are in sorted order i.e. the largest r elements of  $\Sigma$  are in sorted order.  $K_{r,n}$  is obtained by concatenating sublists  $(n-(r-1), n-(r-2), \dots, n)$  and  $(n-r, n-(r+1), \dots, 3, 2, 1)$ . Therefore a permutation  $K_{r,n}$  can be denoted as follows  $(n - (r - 1), n - (r - 2), \dots, n - r, n - (r + 1), \dots, 3, 2, 1).$ Therefore,  $K_{1,n}$  is  $(n, n-1, \ldots, 3, 2, 1)$  which is  $R_n$  and  $K_{n,n}$  $(1,2,\ldots,n-1,n)$  which is  $I_n$ . Let *LE* denote execution of Left-Rotate move followed by a Exchange move and RE denote execution of Right-Rotate move followed by a Exchange move. Further, let  $(LE)^p$  and  $(RE)^p$  be p consecutive executions of RE and RE respectively. Similarly, let  $L^p$  and  $R^p$  be p consecutive executions of L and R respectively.

# A. Background

A Cayley graph  $\Gamma$  defined on Symmetric group  $S_n$ , corresponding to an operation  $\Psi$  with a generator set G has n! vertices each vertex corresponding to a unique permutation. An edge in  $\Gamma$  from a vertex u to another vertex v indicates that there exits a generator  $g \in G$  such that when g is applied to u one obtains v. Applying a generator is called as making a *move*. An upper bound of x moves to sort any permutation in  $S_n$  indicates that the diameter of  $\Gamma$  is at most x. An exact upper bound equals the diameter of  $\Gamma$ . Cayley graphs have many properties that render them apt for computer interconnection networks [4], [5]. Various operations to sort permutations have been posed that are of theoretical and practical interest [5].

Jerrum showed that when the number of generators is greater than one, the computation of minimum length of sequence of generators to sort a permutation is intractable [14]. LRE operation has three generators and the complexity

<sup>&</sup>lt;sup>1</sup>This article is submitted to 10th International Advanced Computing Conference, IACC-2020.

of transforming one permutation in to another with LRE unknown. Exchange move is a reversal of length two, in fact it is a prefix reversal of length two.

For sorting permutations with (unrestricted) prefix reversals the operation that has n-1 generators, the best known upper bound is 18n/11 + O(1) [9]. In LRE operation, both left and right rotate cyclically shifts the entire permutation. In contrast, [12] an extended bubblesort is considered, where an additional swap is allowed between elements in positions 1 and n. We call an operation say  $\Psi$  symmetric if for any generator of  $\Psi$ its *inverse* is also in  $\Psi$ . Exchange operation is inverse of itself whereas left and right rotate are inverses of one another, thus, LRE is symmetric. Both LE and LRE are restrictive compared to the other operations that are studied in the context of genetics e.g. [6]. Research in the area of Cayley graphs has been active. Cayley graphs are studied pertaining to their efficacy in modelling a computer interconnection network, their properties in terms of diameter, presence of greedy cycles in them etc. [11], [13], [15]. Efficient computation of all distances, some theoretical properties of specific Cayley graphs, and efficient counting of groups of permutations in  $S_n$  with related properties have been recently studied [7], [8], [10], [16].

### **II. ALGORITHM LRE**

Algorithm LRE sorts  $R_n$  in stages. It first transforms  $R_n$  which is identical to  $K_{1,n}$  into  $K_{2,n}$  by executing an E move. Subsequently,  $K_{i+1,n}$  is obtained from  $K_{i,n}$  by executing the moves specified by Lemma 1. Thus, eventually we obtain  $K_{n,n}$  which is identical to  $I_n$ . Pseudo Code for the Algorithm LRE is shown below.

## Algorithm LRE

Input:  $R_n$ . Output:  $I_n$ . Initialization:  $\forall i \ \pi[i] = R_n[i]$ . All moves are executed on  $\pi$ .

A	lgori	ithm	1	A	lgor	ithm	LRE
---	-------	------	---	---	------	------	-----

1:	1: for $r \in (1,, n-2)$ do				
2:	if $r = (n-2)$ then				
3:	Execute $R^2$				
4:	Execute E move				
5:	else				
6:	Execute $(L)^{r-1}$				
7:	Execute E move				
8:	Execute $(RE)^{r-1}$				
9:	end if				
10:	10: end for				

# A. Analysis

**Lemma 1.** The number of moves required to obtain  $K_{r+1,n}$ from  $K_{r,n} \forall r \in (1, ..., n-3)$  is 3r-2.

*Proof.* According to the definition,  $K_{r,n}$  is  $(n - (r - 1), n - (r - 2), \dots n - 1, n, n - r, n - (r + 1), n - 1)$  $(r+2),\ldots 3,2,1).$ 

Executing  $L^{r-1}$  on  $K_{r,n}$  yields  $(n, n-r, n-(r+1), n-(r+2), \dots 3, 2, 1, n-(r-1), n-(r-1),$  $(r-2), \ldots n-1).$ An E move is executed to obtain  $(n-r, n, n-(r+1), n-(r+2), \dots 3, 2, 1, n-(r-1), n (r-2), \ldots n-1).$ Finally,  $(RE)^{r-1}$  is executed to obtain  $(n-r, n-(r-1), n-(r-2), \dots n-1, n, n-(r+1), n-($  $(r+2),\ldots,(2,1)$  which is  $K_{r+1,n}$ . Therefore, the total number of moves required to obtain  $K_{r+1,n}$  from  $K_{r,n}$  is (r-1) + 1 + 2(r-1) = 3r - 2.

**Lemma 2.** The number of moves required to obtain  $K_{n,n}$  from  $K_{n-2,n}$  is 3.

*Proof.* According to the definition,  $K_{n-2,n}$  is (3, 4, ..., n - 1, n, 2, 1). Executing  $R^2$  on  $K_{n-2,n}$  yields  $(2, 1, 3, \ldots, n - 1, n)$ . Then executing an E move yields  $(1,2,3\ldots,n-1,n)$  which is  $K_{n,n}$ . Therefore, three moves suffice to transform  $K_{n-2,n}$  into  $K_{n,n}$ . 

Theorem 3. An upper bound for the number of moves required to sort  $R_n$  with LRE is  $\frac{3}{2}n^2$ .

*Proof.* Let J(n) be the number of moves required to sort  $R_n$ with LRE. According to Lemma 1, the number of moves required to obtain  $K_{r+1,n}$  from  $K_{r,n}$  is 3r-2. Let A(n) be the number of moves required to obtain  $K_{n-2,n}$  from  $K_{1,n}$ (which is  $R_n$ ). Then

A

$$(n) = \sum_{r=1}^{n-3} (3r-2)$$
  
=  $3\sum_{r=1}^{n-3} r - (2(n-3))$   
=  $\frac{3}{2}(n-2)(n-3) - 2n + 6$   
=  $\frac{3}{2}(n^2 - 5n + 6) - 2n + 6$   
=  $\frac{3}{2}n^2 - \frac{15}{2}n + 9 - 2n + 6$   
=  $\frac{3}{2}n^2 - \frac{19}{2}n + 15$ 

According to Lemma 2, the number of moves required to obtain  $K_{n,n}$  from  $K_{n-2,n}$  is 3. Therefore,

$$\begin{aligned} U(n) &= A(n) + 3\\ &= \frac{3}{2}n^2 - \frac{19}{2}n + 18 \end{aligned}$$

Therefore, the total number of moves required to sort  $R_n$  with LRE is  $\frac{3}{2}n^2 - \frac{19}{2}n + 18$ . Ignoring the lower order terms an upper bound for number of moves required to sort  $R_n$  with LRE is  $\frac{3n^2}{2}$ . This is the first non-trivial upper bound for the number of moves required to sort  $R_n$  with LRE. 

#### **III. ALGORITHM LRE1**

We designed Algorithm LRE1 in order to obtain the

tighter upper bound for sorting  $R_n$  with LRE. We define a permutation  $K'_{r,n} \in S_n$  as follows. The largest r elements of  $\Sigma$  i.e.  $n - (r-1), n - (r-2), \ldots n$  are in sorted order.  $K'_{r,n}$  is obtained by concatenating sublists  $(n-r, n-(r+1), \ldots 3, 2, 1)$  and  $(n - (r-1), n - (r-2), \ldots n)$ .  $K_{r,n}$  and  $K'_{r,n}$  differ by the starting position of sublist  $(n - (r-1), n - (r-2), \ldots n)$ . The starting position of  $(n - (r-1), n - (r-2), \ldots n)$  in  $K_{r,n}$  is 1 whereas in  $K'_{r,n}$  it is n - r + 1. Algorithm LRE1 first transforms  $R_n$  into  $K'_{\lfloor \frac{n}{2} \rfloor, n}$ . Then it transforms  $K'_{\lfloor \frac{n}{2} \rfloor, n}$  into  $K'_{n,n}$  which is  $I_n$ . Let  $k = \lfloor \frac{n}{2} \rfloor$  and k' = n - k. Let J'(n) be the number of moves executed by Algorithm LRE1 to sort  $R_n$ .

Input:  $R_n$ . Output:  $I_n$ . Initialization:  $\forall i \ \pi[i] = R_n[i]$ .  $k = \lfloor \frac{n}{2} \rfloor$ ,  $k' = n - k = \lceil \frac{n}{2} \rceil$ All moves are executed on  $\pi$ .

# A. Analysis

**Lemma 4.** The permutation obtained after executing D1 and D2 of Algorithm LRE1 is  $(n-1, n-2, ..., \lceil \frac{n}{2} \rceil + 2, n, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil - 1, ..., 3, 2, 1, \lceil \frac{n}{2} \rceil + 1)$  and the number of moves executed is 2n - 6 when n is even and 2n - 8 when n is odd  $\forall n \ge 6$ . *Proof.* Execution of E move on  $R_n$  in D1 yields

 $1 + 4 * \left( \lfloor \frac{n}{2} \rfloor - 2 \right) + 1 = 4 \lfloor \frac{n}{2} \rfloor - 6 = \begin{cases} 2n - 6 & \text{if } n \text{ is even} \\ 2n - 8 & \text{if } n \text{ is odd} \end{cases}$ 

**Lemma 5.** The permutation obtained after D3 and D4 of LRE1 algorithm are executed is  $K_{\lfloor \frac{n}{2} \rfloor,n}'$  and the number of moves executed in the above two steps is  $\frac{3n^2-34n+112}{8}$  when *n* is even and  $\frac{3n^2-40n+149}{8}$  when *n* is odd  $\forall n \geq 8$ .

*Proof.* From Lemma 4, the permutation obtained after steps D1 and D2 is

 $(n-1, n-2, \dots, \lceil \frac{n}{2} \rceil + 2, n, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil - 1, \dots, 3, 2, 1, \lceil \frac{n}{2} \rceil + 1).$ When i = 0 in step D3 only E move is executed and permutation thus obtained is

 $(n-2, n-1, \ldots, \lceil \frac{n}{2} \rceil + 2, n, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil - 1, \ldots, 3, 2, 1, \lceil \frac{n}{2} \rceil + 1).$ When i = 1 in step D3 only L move is executed and permutation thus obtained is

 $(n-1,\ldots,\lceil\frac{n}{2}\rceil+2,n,\lceil\frac{n}{2}\rceil,\lceil\frac{n}{2}\rceil-1,\ldots,3,2,1,\lceil\frac{n}{2}\rceil+1,n-2).$ There after in each iteration in step D3, E move,  $(RE)^{i-1}$  and  $L^i$  are executed so that the elements between  $\pi[1]$  and  $\pi[n-i+2]$  are left rotated. Thus, the permutation obtained after step D3 is  $(n-1,n,\lceil\frac{n}{2}\rceil,\lceil\frac{n}{2}\rceil-1,\ldots,2,1,\lceil\frac{n}{2}\rceil+1,\ldots,n-2)$  Algorithm 2 Algorithm LRE1 1: D1: 2: if  $k \neq 1$  then Execute E move 3: 4: **end if** 5: if  $k \geq 3$  then D2: 6: Execute  $(LE)^{k-2}$ 7: Execute  $(RE)^{k-2}$ 8: Execute L move 9: 10: D3: if  $k \ge 4$  then 11: for  $i \in (0, ..., k - 1)$  do 12: if  $\pi[1] = (n-1)$  then 13: Execute E move 14: if  $i \ge 1$  then 15: Execute  $(RE)^{i-1}$  $16^{-1}$ end if 17: 18: end if Execute  $(L)^i$ 19: end for 20: 21: end if 22: end if 23: D4: 24: if  $k \neq 1$  then Execute  $(L)^2$ 25: else 26: Execute L move 27: 28: end if 29: D5: 30: Execute E move 31: if k' > 3 then D6: 32: Execute  $(LE)^{k'-2}$ 33: Execute  $(RE)^{k'-2}$ 34: 35: if  $k' \geq 4$  then 36: D7: Execute L move 37: 38: **D8**: for  $i \in (0, ..., k' - 1)$  do 39: 40: if  $\pi[1] = (k' - 1)$  then Execute E move 41: if  $i \ge 1$  then 42: Execute  $(RE)^{i-1}$ 43: 44: end if 45: end if if  $i \neq (k'-3)$  then 46: Execute  $(L)^i$ 47: end if 48: end for 49: 50: D9: Execute R move 51: 52: end if 53: end if

and the number of moves executed in each iteration is 1 + 2(i - 1) + i = 3i - 1.

Therefore, the total number of moves executed in step D3 is  $1 + 1 + \sum_{j=2}^{\lfloor \frac{n}{2} \rfloor - 3} (3j - 1)$   $= 2 + \sum_{i=2}^{\lfloor \frac{n}{2} \rfloor - 3} (3i - 1)$   $= \begin{cases} \frac{3n^2 - 34n + 96}{8} & \text{if } n \text{ is even} \\ \frac{3n^2 - 40n + 133}{8} & \text{if } n \text{ is odd} \end{cases}$ Execution of L<sup>2</sup> in step D4 yields

 $(\lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil - 1, \dots, 2, 1, \lceil \frac{n}{2} \rceil + 1, \dots, n - 2, n - 1, n)$  which is  $K_{\lfloor \frac{n}{2} \rfloor,n}$ . Therefore, the total number of moves executed in steps D3 and D4 are  $\begin{cases} \frac{3n^2 - 34n + 112}{8} & \text{if } n \text{ is even} \\ \frac{3n^2 - 40n + 149}{8} & \text{if } n \text{ is odd} \end{cases}$ . 

Lemma 6. The permutation obtained after executing D5 and D6 of LRE1 algorithm is

 $\left(1, \left\lceil \frac{n}{2} \right\rceil - 1, \left\lceil \frac{n}{2} \right\rceil - 2, \dots, 2, \left\lceil \frac{n}{2} \right\rceil, \left\lceil \frac{n}{2} \right\rceil + 1, \left\lceil \frac{n}{2} \right\rceil + 2, \dots, n-1, n\right)$ and the number of moves executed in the above two steps is 2n-7 when n is even and 2n-5 when n is odd  $\forall n > 5$ .

*Proof.* According to Lemma 5, the permutation obtained after the steps D1 to D4 is

 $\left(\left\lceil \frac{n}{2}\right\rceil, \left\lceil \frac{n}{2}\right\rceil - 1, \left\lceil \frac{n}{2}\right\rceil - 2, \dots, 2, 1, \left\lceil \frac{n}{2}\right\rceil + 1, \left\lceil \frac{n}{2}\right\rceil + 2, \dots, n-1, n\right).$ Now, executing E move in step D5 yields

$$\left(\left\lceil \frac{n}{2}\right\rceil - 1, \left\lceil \frac{n}{2}\right\rceil, \left\lceil \frac{n}{2}\right\rceil - 2, \dots, 2, 1, \left\lceil \frac{n}{2}\right\rceil + 1, \left\lceil \frac{n}{2}\right\rceil + 2, \dots, n-1, n\right)$$
  
Then executing  $(LE)^{k'-2}$  in step D6 yields

 $(1, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil + 1, \lceil \frac{n}{2} \rceil + 2, \dots, n-1, n, \lceil \frac{n}{2} \rceil - 1, \lceil \frac{n}{2} \rceil - 2, \dots, 2).$ Then executing  $(RE)^{k'-2}$  in step *D*6 yields

 $(1, \lceil \frac{n}{2} \rceil - 1, \lceil \frac{n}{2} \rceil - 2, \dots, 2, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil + 1, \lceil \frac{n}{2} \rceil + 2, \dots, n-1, n).$ Therefore, the total number of moves executed in steps D5 and D6 is

$$1 + 4(k' - 2) = 4k' - 7 = \begin{cases} 2n - 7 & \text{if } n \text{ is even} \\ 2n - 5 & \text{if } n \text{ is odd} \end{cases}.$$

Lemma 7. The permutation obtained after executing D7 and D8 of LRE1 algorithm is  $(2, 3, \ldots, n-1, n, 1)$  and the number of moves executed in the above two steps is  $\frac{3n^2-38n+128}{8}$  when n is even and  $\frac{3n^2-32n+93}{8}$  when n is odd  $\forall n \ge 7$ .

Proof. According to Lemma 6, the permutation obtained after steps D1 to D6 is

 $(1, \lceil \frac{n}{2} \rceil - 1, \lceil \frac{n}{2} \rceil - 2, \dots, 2, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil + 1, \lceil \frac{n}{2} \rceil + 2, \dots, n-1, n).$ L move is executed in step D7 and the permutation thus obtained is

 $\left(\left\lceil \frac{n}{2}\right\rceil - 1, \left\lceil \frac{n}{2}\right\rceil - 2, \dots, 2, \left\lceil \frac{n}{2}\right\rceil, \left\lceil \frac{n}{2}\right\rceil + 1, \left\lceil \frac{n}{2}\right\rceil + 2, \dots, n-1, n, 1\right).$ When i = 0 in step D8, only E move is executed and the permutation thus obtained is

 $(\lceil \frac{n}{2}\rceil - 2, \lceil \frac{n}{2}\rceil - 1, \dots, 2, \lceil \frac{n}{2}\rceil, \lceil \frac{n}{2}\rceil + 1, \lceil \frac{n}{2}\rceil + 2, \dots, n-1, n, 1).$ When i = 1 in step D8, only L move is executed and the permutation thus obtained is

 $\left(\left\lceil \frac{n}{2}\right\rceil - 1, \ldots, 2, \left\lceil \frac{n}{2}\right\rceil, \left\lceil \frac{n}{2}\right\rceil + 1, \left\lceil \frac{n}{2}\right\rceil + 2, \ldots, n-1, n, 1, \left\lceil \frac{n}{2}\right\rceil - 2\right).$ There after in each iteration in step D8 except when i =(k'-3), E move,  $(RE)^{i-1}$  and  $L^i$  are executed so that the elements between  $\pi[1]$  and  $\pi[n-i+2]$  are left rotated. When i = k' - 3 only E move and  $(RE)^{i-1}$  are executed. Thus, the obtained permutation after step D8 is  $(2, 3, \ldots, n-1, n, 1)$ . The number of moves executed in step D8 is

$$1 + 1 + 1 + \sum_{\substack{j=2\\ j=2}}^{\lfloor \frac{n}{2} - 4 \rfloor} (3j - 1) + 1 + 2 * \lfloor \frac{n}{2} - 4 \rfloor$$
  
=  $4 + \sum_{\substack{j=2\\ g=2}}^{\lfloor \frac{n}{2} - 4 \rfloor} (3j - 1) + 2 * \lfloor \frac{n}{2} - 4 \rfloor$   
=  $\begin{cases} \frac{3n^2 - 38n + 128}{8} & \text{if } n \text{ is even} \\ \frac{3n^2 - 32n + 93}{8} & \text{if } n \text{ is odd} \end{cases}$ 

**Lemma 8.** Algorithm LRE1 is correct.

*Proof.* According to Lemma 7, the permutation obtained after steps D1 to D8 is

 $(2,3,\ldots,\lceil\frac{n}{2}\rceil-1,\lceil\frac{n}{2}\rceil,\lceil\frac{n}{2}\rceil+1,\lceil\frac{n}{2}\rceil+2,\ldots,n-1,n,1).$ Executing R move in step D9 yields  $(1, 2, 3, \dots, \lceil \frac{n}{2} \rceil - 1, \lceil \frac{n}{2} \rceil, \lceil \frac{n}{2} \rceil + 1, \lceil \frac{n}{2} \rceil + 2, \dots, n-1, n)$  which is  $I_n$ . Hence proves the lemma.

**Theorem 9.** The number of moves required to sort  $R_n$  with LRE1 algorithm is

$$J'(n) = \begin{cases} 2 & \text{if } n = 3 \\ 4 & \text{if } n = 4 \\ 8 & \text{if } n = 5 \\ 13 & \text{if } n = 6 \\ 20 & \text{if } n = 7 \\ \frac{3n^2 - 20n + 72}{4} & \text{if } n \ge 8 \text{ and } n \text{ is even} \\ \frac{3n^2 - 20n + 73}{4} & \text{if } n \ge 8 \text{ and } n \text{ is odd} \end{cases}$$

Proof. Case-i: n is 3

When n=3, the values of k and k' are 1 and 2 respectively. So, only steps D1 and D4 are executed. Therefore, the number of moves executed are 1 + 1 = 2.

Case-ii: n is 4

When n=4, the values of both k and k' is 2, So only steps D1, D4 and D5 are executed. Therefore, the total number of moves executed are 1 + 2 + 1 = 4.

Case-iii: n is 5

When n=5, the values of k and k' are 2 and 3 respectively. So only steps D1, D4, D5 and D6 are executed. According to Lemma 6, the number of moves executed by steps D5 and D6 is 2n-5 when n is odd. Therefore, the total number of moves executed are 1+2+2n-5=2n-2=8.

Case-iv: n is 6

When n=6, the values of both k and k' is 3. Therefore steps D1, D2, D4, D5 and D6 are executed. According to Lemma 4, the number of moves executed by steps D1and D2 is 2n - 6 when n is even. According to Lemma 6, the number of moves executed by steps D5 and D6 is 2n - 7 when n is even. Therefore the total number of moves 2n-7 when n is even. Therefore, the total number of moves executed are 2n-6+2+2n-7=4n-11=13.

Case-v: n is 7

When n=7, the values of k and k' are 3 and 4 respectively. Therefore steps D1, D2, D4, D5 and D6 are executed. According to Lemma 4, the number of moves executed by steps D1 and D2 is 2n - 8 when n is odd. The number of moves executed by step D4 is 2. According to Lemma 6, the number of moves executed by steps D5

and D6 is 2n - 5 when n is odd. According to Lemma 7, the number of moves executed by steps D7 and D8 is  $\frac{3n^2-32n+93}{8}$  when n is odd. Number of moves executed by step D9 is 1. Therefore, the total number of moves executed is  $2n - 8 + 2 + 2n - 5 + \frac{3n^2-32n+93}{8} + 1 = 4n - 11 + \frac{3n^2-32n+93}{8} + 1 = 20.$ 

Case-vi:  $n \ge 8$  and n is even

In this case all steps from D1 to D9 are executed. According to Lemma 4, the number of moves executed by steps D1 and D2 is 2n - 6 when n is even. According to Lemma 5, the number of moves executed by steps D3 and D4 is  $\frac{3n^2-34n+112}{8}$  when n is even. According to Lemma 6, the number of moves executed by steps D5 and D6 is 2n - 7 when n is even. According to Lemma 7, the number of moves executed by steps D7 and D8 is  $\frac{3n^2-38n+128}{8}$ when n is even. Number of moves executed by step D9 is 1. Therefore, the total number of moves executed by Algorithm LRE1 is

$$J'(n) = 2n - 6 + \frac{3n^2 - 34n + 112}{8} + 2n - 7 + \frac{3n^2 - 38n + 128}{8} + 1$$
$$= \frac{3n^2 - 20n + 72}{4}$$

Case-vii:  $n \ge 8$  and n is odd

In this case all steps from D1 to D9 are executed. According to Lemma 4, the number of moves executed by steps D1 and D2 is 2n - 8 when n is odd. According to Lemma 5, the number of moves executed by steps D3 and D4 is  $\frac{3n^2 - 40n + 149}{8}$  when n is odd. According to Lemma 6, the number of moves executed by steps D5 and D6 is 2n - 5when n is odd. According to Lemma 7, the number of moves executed by steps D7 and D8 is  $\frac{3n^2 - 32n + 93}{8}$  when n is odd. Number of moves executed by step D9 is 1. Therefore, the total number of moves executed by Algorithm LRE1 is

$$J'(n) = 2n - 8 + \frac{3n^2 - 40n + 149}{8} + 2n - 5 + \frac{3n^2 - 32n + 93}{8} + 1$$
$$= \frac{3n^2 - 20n + 73}{4}$$

Therefore, ignoring the lower order terms the new tighter upper bound for number of moves required to sort  $R_n$  with LRE is  $\frac{3n^2}{4}$ .

## **IV. EXHAUSTIVE SEARCH RESULTS**

A branch and bound algorithm that employs BFS, i.e. Algorithm Search, has been designed for computing the minimum number of moves to sort  $R_n$  for a given n. It yielded values of 43 for n = 10 and 53 for n = 11. Thus, including the current values, the identified minimum number of moves for for n = 1...11 are respectively (0, 1, 2, 4, 8, 13, 19, 26, 34, 43, 53). A list of permutations whose distance has been computed is maintained and the execution in every branch terminates either upon reaching  $I_n$  or exceeding a bound. E, L and R generators are applied to each of the intermediate permutations yielding the corresponding permutations. We avoid application of two successive generators that are inverses of each other as such a sequence cannot be a part of optimum solution. Notation: Node contains a permutation  $\in S_n$  and its distance from  $R_n$  corresponds to the minimum number of moves. With this algorithm and better computational resources one will be able to compute the corresponding values for larger values of n.

# **Algorithm Search**

Initialization: The source vertex  $\delta$  contains the permutation  $R_n$  and its path is initialized to null. It is enqueued into BFS queue Q.

Input: $R_n$ . Output: Optimum number of moves to reach  $I_n$  with LRE.

with Di					
Algorit	hm 3 Algorithm Search				
1: <b>wh</b>	ile (Q is not empty) do				
2: I	Dequeue $u$ from $Q$				
3: <b>i</b> t	f ( $u$ is visited) then				
4:	continue				
5: <b>e</b>	nd if				
6: N	Mark $u$ as visited				
7: <b>i</b>	if $(u \text{ is } I_n)$ then				
8:	{Array is sorted}				
9:	return length of <i>u.path</i>				
10:	break				
11: <b>e</b>	nd if				
12: <b>i</b>	f (Last move on $u.path \neq E$ or $u.path = null$ ) then				
13:	Execute E on $u \to v$				
14:	if $v$ is not visited then				
15:	$v.path \leftarrow u.path$ followed by E				
16:	Enqueue $v$ to Q				
17:	end if				
18: <b>e</b>	nd if				
19: <b>i</b>	f (Last move on $u.path \neq L$ or $u.path = null$ ) then				
20:	Execute R on $u \to v$				
21:	if $v$ is not visited then				
22:	$v.path \leftarrow u.path$ followed by R				
23:	Enqueue $v$ to Q				
24:	end if				
25: <b>e</b>	nd if				
26: <b>i</b>	f (Last move on $u.path \neq R$ or $u.path = null$ ) then				
27:	Execute L on $u \to v$				
28:	if $v$ is not visited then				
29:	$v.path \leftarrow u.path$ followed by L				
30:	Enqueue $v$ to Q				
31:	end if				
33: <b>end</b>	l while				

## V. RESULTS

Comparison of the number of moves required to sort  $R_n$ with LRE by various algorithms. The first column shows n, the size of permutation. Subsequent columns show the number of moves required to sort  $R_n$  with Algorithms LRE, LRE1and Search respectively.

# CONCLUSION

The first known upper bound for sorting  $R_n$  with LRE has been shown. A tighter upper bound has been derived. The

n	LRE	LRE1	Search(Optimal)
3	3	2	2
4	4	4	4
5	8	8	8
6	15	13	13
7	25	20	19
8	38	26	26
9	38 54	34	34
10	73	43 54	43
11	95	54	53

future work consists of identifying the exact upper bound for sorting  $R_n$  with LRE. The identification of the diameter of the LRE Cayley graph and the characterization of permutations that are farthest from  $I_n$  in this Cayley graph are open questions.

#### ACKNOWLEDGEMENT

Venkata Vyshnavi Ravella and CK Phani Datta have been contributing towards implementation, testing and analysis of the algorithms. Authors thank Jayakumar P for helpful suggestions. Authors thank Tony Bartoletti for personal communication.

### REFERENCES

- [1] The On-Line Encyclopedia of Integer Sequences: oeis.org.
- [2] Kuppilil S. S., Chitturil, B., and Srinath T., "An Upper Bound for Sorting R<sub>n</sub> with LE," *ICACDS 2019*.
- [3] Kuppili1 S. S., Chitturi1, B., "Exact Upper Bound for Sorting R<sub>n</sub> with LE," Discrete Math Algorithms Appl. 2020, doi.org/10.1142/S1793830920500330.
- [4] Akers, S. B., Krishnamurthy, B., "A group-theoretic model for symmetric interconnection networks," *IEEE Transactions on Computers*, 38(4):555566, 1989.
- [5] Lakshmivarahan, S., Jho, J-S., and Dhall, S. K., "Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey," *Parallel Computing*, 19:361407, 1993.
- [6] Chitturi, B., "A note on complexity of genetic mutations," *Discrete Mathematics, Algorithms and Applications,* 3, 269-286, 2011.
- [7] Chitturi, B., and Das, P., "Sorting permutations with transpositions in  $O(n^3)$  amortized time," *Theoretical Computer Science*, 2018.
- [8] Biniaz, A., Jain, K., Lubiw, A., Masárová Z., Miltzow, T., Mondal, D., Naredla, A.M., Tkadlec, J., and Turcotte, A., "Token Swapping on Trees," http://arxiv.org/abs/1903.06981, 2019.
- [9] Chitturi, B., Fahle, W., Meng, Z., Morales, L., Shields, C. O., Sudborough H., "An (18/11) n upper bound for sorting by prefix reversals," *Theoretical Computer Science*, 410(36): 3372-3390, 2009.
- [10] Chitturi, B., Computing cardinalities of subsets of S<sub>n</sub> with k adjacencies, JCMCC, (to appear) 2020.
- [11] Erskine, G., and James T., "Large Cayley graphs of small diameter," Discrete Applied Mathematics, 2018.
- [12] Feng, X., Chitturi, B., Sudborough, H., "Sorting circular permutations by bounded transpositions,", *Advances in Computational Biology*, Springer, New York, NY (2010), pp. 725-736.
- [13] Gostevsky, D.A. and Konstantinova, E.V., "Greedy cycles in the star graphs," *Discrete mathematics and mathematical cybernetics*, 15(0):205-213, 2018.
- [14] Jerrum, M. R., "The complexity of finding minimum-length generator sequences," *Theoretical Computer Science*, 36:265-289, 1985.
- [15] Mokhtar, H., "A few families of Cayley graphs and their efficiency as communication networks," *Bulletin of the Australian Mathematical Society*, 95, no. 3, 518-520, 2017.
- [16] Zhang, T., and Gennian G., "Improved lower bounds on the degreediameter problem," *Journal of Algebraic Combinatorics*, 1-12, 2018.