# An Upper Bound for Sorting $R_{n}$ with LRE 

Sai Satwik Kuppili* and Bhadrachalam Chitturi ${ }^{\dagger}$<br>* Dept. of Computer Science and Engineering, Amrita University<br>Amritapuri, India<br>${ }^{\dagger}$ Dept. of Computer Science, University of Texas at Dallas<br>Texas, USA<br>*VMware Inc. Pune, India<br>Email: *satwik.kuppili@gmail.com, ${ }^{\dagger}$ chalam@utdallas.edu


#### Abstract

${ }^{1}$ 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 $\Sigma . I_{n}=(1,2,3, \ldots, n)$ 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 $L R E$ operation, 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 $L R E$ operation are known for $n \leq 11$ 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 $L R E$ 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 $L R E$; (b) a tighter upper bound for the number of moves required to sort $R_{n}$ with $L R E$; 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 $L R E$ 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 $(1,2, \ldots, n)$. [1]." We call this operation as $L R E . L R E$ 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 $E$ respectively. $R_{n}$ denotes $(n, n-1, \ldots, 2,1)$ whereas $I_{n}$ denotes the sorted order or identity permutation: $(1,2, \ldots, n)$. Sorting a permutation $\pi$ in this article refers to transforming $\pi$ into $I_{n}$ with $L R E$ operation. The alphabet is $\Sigma=(1,2,3, \ldots, n)$. [2], [3] studied a more restricted version of this problem, i.e. $L E$ operation where the operation $R$ is disallowed and appears in

[^0]OEIS with sequence number A048200 [1]. We note that the results of [2], [3] are applicable to $R E$ 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 $L R E$ into $I_{n}$.
The optimum number of moves to sort $R_{n}$ with $L R E$ 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 $n-(r-1), n-(r-2), \ldots n$ are in sorted order i.e. the largest $r$ elements of $\Sigma$ are in sorted order. $K_{r, n}$ is obtained by concatenating sublists $(n-(r-1), n-(r-2), \ldots n)$ and $(n-r, n-(r+1), \ldots 3,2,1)$. Therefore a permutation $K_{r, n}$ can be denoted as follows $(n-(r-1), n-(r-2), \ldots n, n-r, n-(r+1), \ldots 3,2,1)$. 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 $L E$ denote execution of Left-Rotate move followed by a Exchange move and $R E$ denote execution of Right-Rotate move followed by a Exchange move. Further, let $(L E)^{p}$ and $(R E)^{p}$ be $p$ consecutive executions of $R E$ and $R E$ 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
of transforming one permutation in to another with $L R E$ 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 $18 n / 11+O(1)$ [9]. In $L R E$ 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, $L R E$ is symmetric. Both $L E$ and $L R E$ 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 $L R E$ 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 $L R E$ is shown below.

## Algorithm LRE

Input: $R_{n}$. Output: $I_{n}$.
Initialization: $\forall i \pi[i]=R_{n}[i]$.
All moves are executed on $\pi$.

```
Algorithm 1 Algorithm LRE
    for \(r \in(1, \ldots, n-2)\) do
        if \(r=(n-2)\) then
            Execute \(R^{2}\)
            Execute E move
        else
            Execute \((L)^{r-1}\)
            Execute E move
            Execute \((R E)^{r-1}\)
        end if
    end for
```


## A. Analysis

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

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

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
$(3,4, \ldots, n-1, n, 2,1)$. 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 $L R E$ is $\frac{3}{2} n^{2}$.
Proof. Let $J(n)$ be the number of moves required to sort $R_{n}$ with $L R E$. According to Lemma 1, the number of moves required to obtain $K_{r+1, n}$ from $K_{r, n}$ is $3 r-2$. Let $A(n)$ be the number of moves required to obtain $K_{n-2, n}$ from $K_{1, n}$ (which is $R_{n}$ ). Then

$$
\begin{aligned}
A(n) & =\sum_{r=1}^{n-3}(3 r-2) \\
& =3 \sum_{r=1}^{n-3} r-(2(n-3)) \\
& =\frac{3}{2}(n-2)(n-3)-2 n+6 \\
& =\frac{3}{2}\left(n^{2}-5 n+6\right)-2 n+6 \\
& =\frac{3}{2} n^{2}-\frac{15}{2} n+9-2 n+6 \\
& =\frac{3}{2} n^{2}-\frac{19}{2} n+15
\end{aligned}
$$

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

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

Therefore, the total number of moves required to sort $R_{n}$ with $L R E$ 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 $L R E$ is $\frac{3 n^{2}}{2}$. This is the first non-trivial upper bound for the number of moves required to sort $R_{n}$ with $L R E$.

## III. Algorithm LRE1

We designed Algorithm $L R E 1$ in order to obtain the
tighter upper bound for sorting $R_{n}$ with $L R E$. We define a permutation $K_{r, n}^{\prime} \in S_{n}$ as follows. The largest $r$ elements of $\Sigma$ i.e. $n-(r-1), n-(r-2), \ldots n$ are in sorted order. $K_{r, n}^{\prime}$ 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 $K_{r, n}^{\prime}$ 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 $K_{r, n}^{\prime}$ it is $n-r+1$. Algorithm LRE1 first transforms $R_{n}$ into $K_{\left\lfloor\frac{n}{2}\right\rfloor, n}^{\prime}$. Then it transforms $K_{\left\lfloor\frac{n}{2}\right\rfloor, n}^{\prime}$ into $K_{n, n}^{\prime}$ which is $I_{n}$. Let $k=\left\lfloor\frac{n}{2}\right\rfloor$ and $k^{\prime}=n-k$. Let $J^{\prime}(n)$ be the number of moves executed by Algorithm LRE1 to sort $R_{n}$.

Input: $R_{n}$. Output: $I_{n}$.
Initialization: $\forall i \pi[i]=R_{n}[i] . k=\left\lfloor\frac{n}{2}\right\rfloor, k^{\prime}=n-k=\left\lceil\frac{n}{2}\right\rceil$
All moves are executed on $\pi$.

## A. Analysis

Lemma 4. The permutation obtained after executing D1 and D2 of Algorithm LRE1 is
$\left(n-1, n-2, \ldots,\left\lceil\frac{n}{2}\right\rceil+2, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1,\left\lceil\frac{n}{2}\right\rceil+1\right)$ and the number of moves executed is $2 n-6$ when $n$ is even and $2 n-8$ when $n$ is odd $\forall n \geq 6$.
Proof. Execution of E move on $R_{n}$ in $D 1$ yields
$(n-1, n, n-2, \ldots, 3,2,1)$.
Then executing $(L E)^{k-2}$ in $D 2$ yields
$\left(\left\lceil\frac{n}{2}\right\rceil+1, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1, n-1, n-2, \ldots,\left\lceil\frac{n}{2}\right\rceil+2\right)$.
Then executing $(R E)^{k-2}$ in $D 2$ yields
$\left(\left\lceil\frac{n}{2}\right\rceil+1, n-1, n-2, \ldots,\left\lceil\frac{n}{2}\right\rceil+2, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1\right)$.
Then performing $L$ in $D 2$ move yields
$\left(n-1, n-2, \ldots,\left\lceil\frac{n}{2}\right\rceil+2, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1,\left\lceil\frac{n}{2}\right\rceil+1\right)$.
Therefore, the total number of moves executed in step $D 1$ and $D 2$ is
$1+4 *\left(\left\lfloor\frac{n}{2}\right\rfloor-2\right)+1=4\left\lfloor\frac{n}{2}\right\rfloor-6=\left\{\begin{array}{ll}2 n-6 & \text { if } n \text { is even } \\ 2 n-8 & \text { if } n \text { is odd }\end{array}\right.$.

Lemma 5. The permutation obtained after D3 and D4 of LRE1 algorithm are executed is $K_{\left\lfloor\frac{n}{2}\right\rfloor, n}^{\prime}$ and the number of moves executed in the above two steps is $\frac{3 n^{2}-34 n+112}{8}$ when $n$ is even and $\frac{3 n^{2}-40 n+149}{8}$ when $n$ is odd $\forall n \geq 8$.
Proof. From Lemma 4, the permutation obtained after steps $D 1$ and $D 2$ is
$\left(n-1, n-2, \ldots,\left\lceil\frac{n}{2}\right\rceil+2, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1,\left\lceil\frac{n}{2}\right\rceil+1\right)$. When $i=0$ in step $D 3$ only $E$ move is executed and permutation thus obtained is
$\left(n-2, n-1, \ldots,\left\lceil\frac{n}{2}\right\rceil+2, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1,\left\lceil\frac{n}{2}\right\rceil+1\right)$. When $i=1$ in step $D 3$ only $L$ move is executed and permutation thus obtained is
$\left(n-1, \ldots,\left\lceil\frac{n}{2}\right\rceil+2, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1, \ldots 3,2,1,\left\lceil\frac{n}{2}\right\rceil+1, n-2\right)$. There after in each iteration in step $D 3, E$ move, $(R E)^{i-1}$ 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 $D 3$ is $\left(n-1, n,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1 \ldots, 2,1,\left\lceil\frac{n}{2}\right\rceil+1, \ldots, n-2\right)$

```
Algorithm 2 Algorithm LRE1
    D1:
    if }k\not=1\mathrm{ then
        Execute E move
    end if
    if }k\geq3\mathrm{ then
        D2:
        Execute (LE)
        Execute (RE)
        Execute L move
        D3:
        if }k\geq4\mathrm{ then
            for}i\in(0,\ldots,k-1) d
                if }\pi[1]=(n-1)\mathrm{ then
                    Execute E move
                    if}i\geq1\mathrm{ then
                        Execute (RE)}\mp@subsup{)}{}{i-1
                    end if
                    end if
                Execute (L)
            end for
        end if
    end if
    D4:
    if }k\not=1\mathrm{ then
        Execute (L)
    else
        Execute L move
    end if
    D5:
    Execute E move
    if }\mp@subsup{k}{}{\prime}\geq3\mathrm{ then
        D6:
```



```
        Execute (RE) 尔-2
        if }\mp@subsup{k}{}{\prime}\geq4\mathrm{ then
            D7:
            Execute L move
            D8:
            for i\in(0,\ldots,k' - 1) do
                    if }\pi[1]=(\mp@subsup{k}{}{\prime}-1)\mathrm{ then
                Execute E move
                if }i\geq1\mathrm{ then
                        Execute (RE)}\mp@subsup{)}{}{i-1
                end if
            end if
            if }i\not=(\mp@subsup{k}{}{\prime}-3)\mathrm{ then
                Execute (L)}\mp@subsup{}{}{i
            end if
            end for
            D9:
            Execute R move
        end if
    end if
```

and the number of moves executed in each iteration is $1+2(i-1)+i=3 i-1$.
Therefore, the total number of moves executed in step $D 3$ is
$1+1+\sum_{j=2}^{\left\lfloor\frac{n}{2}\right\rfloor-3}(3 j-1)$
$=2+\sum_{i=2}^{\left\lfloor\frac{n}{2}\right\rfloor-3}(3 i-1)$
$= \begin{cases}\frac{3 n^{2}-34 n+96}{8} & \text { if } n \text { is even } \\ \frac{3 n^{2}-40 n+133}{8} & \text { if } n \text { is odd }\end{cases}$
Execution of $L^{2}$ in step $D 4$ yields
$\left(\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1 \ldots, 2,1,\left\lceil\frac{n}{2}\right\rceil+1, \ldots, n-2, n-1, n\right)$ which
is $K_{\left\lfloor\frac{n}{2}\right\rfloor, n}^{\prime}$. Therefore, the total number of moves executed in
steps $D 3$ and $D 4$ are $\left\{\begin{array}{ll}\frac{3 n^{2}-34 n+112}{8} & \text { if } n \text { is even } \\ \frac{3 n^{2}-40 n+149}{8} & \text { if } n \text { is odd }\end{array}\right.$.

Lemma 6. The permutation obtained after executing D5 and D6 of LRE1 algorithm is
$\left(1,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n\right)$ and the number of moves executed in the above two steps is $2 n-7$ when $n$ is even and $2 n-5$ when $n$ is odd $\forall n \geq 5$.
Proof. According to Lemma [5, the permutation obtained after the steps $D 1$ to $D 4$ is
$\left(\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2,1,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n\right)$. Now, executing $E$ move in step $D 5$ yields
$\left(\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2,1,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n\right)$. Then executing $(L E)^{k^{\prime}-2}$ in step $D 6$ yields
$\left(1,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2\right)$.
Then executing $(R E)^{k^{\prime}-2}$ in step $D 6$ yields
$\left(1,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n\right)$.
Therefore, the total number of moves executed in steps $D 5$ and $D 6$ is
$1+4\left(k^{\prime}-2\right)=4 k^{\prime}-7=\left\{\begin{array}{ll}2 n-7 & \text { if } n \text { is even } \\ 2 n-5 & \text { if } n \text { is odd }\end{array}\right.$.

Lemma 7. The permutation obtained after executing $D 7$ 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 $\frac{3 n^{2}-38 n+128}{8}$ when $n$ is even and $\frac{3 n^{2}-32 n+93}{8}$ when $n$ is odd $\forall n \geq 7$.

Proof. According to Lemma6, the permutation obtained after steps $D 1$ to $D 6$ is
$\left(1,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n\right)$. $L$ move is executed in step $D 7$ and the permutation thus obtained is
$\left(\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil-2, \ldots, 2,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n, 1\right)$. When $i=0$ in step D8, only $E$ move is executed and the permutation thus obtained is
$\left(\left\lceil\frac{n}{2}\right\rceil-2,\left\lceil\frac{n}{2}\right\rceil-1, \ldots, 2,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n, 1\right)$. 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 $D 8$ except when $i=$ $\left(k^{\prime}-3\right), E$ move, $(R E)^{i-1}$ and $L^{i}$ are executed so that the elements between $\pi[1]$ and $\pi[n-i+2]$ are left rotated. When $i=k^{\prime}-3$ only $E$ move and $(R E)^{i-1}$ are executed. Thus, the
obtained permutation after step $D 8$ is $(2,3, \ldots, n-1, n, 1)$. The number of moves executed in step $D 8$ is
$1+1+1+\sum_{j=2}^{\left\lceil\frac{n}{2}-4\right\rceil}(3 j-1)+1+2 *\left\lceil\frac{n}{2}-4\right\rceil$
$=4+\sum_{j=2}^{\left\lceil\frac{n}{2}-4\right\rceil}(3 j-1)+2 *\left\lceil\frac{n}{2}-4\right\rceil$
$=\left\{\begin{array}{ll}\frac{3 n^{2}-38 n+128}{8} & \text { if } n \text { is even } \\ \frac{3 n^{2}-32 n+93}{8} & \text { if } n \text { is odd }\end{array}\right.$.

Lemma 8. Algorithm LRE1 is correct.
Proof. According to Lemma 7 the permutation obtained after steps $D 1$ to $D 8$ is
$\left(2,3, \ldots,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n, 1\right)$.
Executing $R$ move in step $D 9$ yields
$\left(1,2,3, \ldots,\left\lceil\frac{n}{2}\right\rceil-1,\left\lceil\frac{n}{2}\right\rceil,\left\lceil\frac{n}{2}\right\rceil+1,\left\lceil\frac{n}{2}\right\rceil+2, \ldots, n-1, n\right)$ which is $I_{n}$. Hence proves the lemma.

Theorem 9. The number of moves required to sort $R_{n}$ with LRE1 algorithm is
$J^{\prime}(n)=\left\{\begin{array}{ll}2 & \text { if } n=3 \\ 4 & \text { if } n=4 \\ 8 & \text { if } n=5 \\ 13 & \text { if } n=6 \\ 20 & \text { if } n=7 \\ \frac{3 n^{2}-20 n+72}{4} & \text { if } n \geq 8 \text { and } n \text { is even } \\ \frac{3 n^{2}-20 n+73}{4} & \text { if } n \geq 8 \text { and } n \text { is odd }\end{array}\right.$.
Proof. Case-i: $n$ is 3
When $n=3$, the values of $k$ and $k^{\prime}$ are 1 and 2 respectively. So, only steps $D 1$ and $D 4$ 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 $k^{\prime}$ is 2 , So only steps $D 1, D 4$ and $D 5$ 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 $k^{\prime}$ are 2 and 3 respectively. So only steps $D 1, D 4, D 5$ and $D 6$ are executed. According to Lemma 6, the number of moves executed by steps $D 5$ and $D 6$ is $2 n-5$ when $n$ is odd. Therefore, the total number of moves executed are $1+2+2 n-5=2 n-2=8$.

Case-iv: $n$ is 6
When $n=6$, the values of both $k$ and $k^{\prime}$ is 3 . Therefore steps $D 1, D 2, D 4, D 5$ and $D 6$ are executed. According to Lemma 4, the number of moves executed by steps $D 1$ and $D 2$ is $2 n-6$ when $n$ is even. According to Lemma 6. the number of moves executed by steps $D 5$ and $D 6$ is $2 n-7$ when $n$ is even. Therefore, the total number of moves executed are $2 n-6+2+2 n-7=4 n-11=13$.
Case-v: $n$ is 7
When $n=7$, the values of $k$ and $k^{\prime}$ are 3 and 4 respectively. Therefore steps $D 1, D 2, D 4, D 5$ and $D 6$ are executed. According to Lemma 4 , the number of moves executed by steps $D 1$ and $D 2$ is $2 n-8$ when $n$ is odd. The number of moves executed by step $D 4$ is 2. According to Lemma 6, the number of moves executed by steps $D 5$
and $D 6$ is $2 n-5$ when $n$ is odd. According to Lemma 7 the number of moves executed by steps $D 7$ and $D 8$ is $\frac{3 n^{2}-32 n+93}{8}$ when $n$ is odd. Number of moves executed by step $D 9$ is 1 . Therefore, the total number of moves executed is $2 n-8+2+2 n-5+\frac{3 n^{2}-32 n+93}{8}+1=$ $4 n-11+\frac{3 n^{2}-32 n+93}{8}+1=20$.

Case-vi: $n \geq 8$ and $n$ is even
In this case all steps from $D 1$ to $D 9$ are executed. According to Lemma 4 the number of moves executed by steps $D 1$ and $D 2$ is $2 n-6$ when $n$ is even. According to Lemma 51 the number of moves executed by steps $D 3$ and $D 4$ is $\frac{3 n^{2}-34 n+112}{8}$ when $n$ is even. According to Lemma 6, the number of moves executed by steps $D 5$ and $D 6$ is $2 n-7$ when $n$ is even. According to Lemma 7, the number of moves executed by steps $D 7$ and $D 8$ is $\frac{3 n^{2}-38 n+128}{8}$ when $n$ is even. Number of moves executed by step $D 9$ is 1 . Therefore, the total number of moves executed by Algorithm $L R E 1$ is

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

Case-vii: $n \geq 8$ and $n$ is odd
In this case all steps from $D 1$ to $D 9$ are executed. According to Lemma 4 , the number of moves executed by steps $D 1$ and $D 2$ is $2 n-8$ when $n$ is odd. According to Lemma 51 the number of moves executed by steps $D 3$ and $D 4$ is $\frac{3 n^{2}-40 n+149}{8}$ when $n$ is odd. According to Lemma 6 , the number of moves executed by steps $D 5$ and $D 6$ is $2 n-5$ when $n$ is odd. According to Lemma 7 , the number of moves executed by steps $D 7$ and $D 8$ is $\frac{3 n^{2}-32 n+93}{8}$ when $n$ is odd. Number of moves executed by step $D 9$ is 1 . Therefore, the total number of moves executed by Algorithm LRE1 is

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

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

## 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 $n=1 \ldots 11$ 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. $\mathrm{E}, \mathrm{L}$ 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 $L R E$.

```
Algorithm 3 Algorithm Search
    while (Q is not empty) do
        Dequeue \(u\) from \(Q\)
        if ( \(u\) is visited) then
            continue
        end if
        Mark \(u\) as visited
        if \(\left(u\right.\) is \(\left.I_{n}\right)\) then
            \{Array is sorted \}
            return length of u.path
            break
        end if
        if (Last move on \(u\).path \(\neq E\) or u.path \(=\) null ) then
            Execute E on \(u \rightarrow v\)
            if \(v\) is not visited then
                \(v . p a t h \leftarrow u\).path followed by E
                Enqueue \(v\) to \(\mathbf{Q}\)
            end if
        end if
        if (Last move on u.path \(\neq L\) or u.path \(=\) null \()\) then
            Execute R on \(u \rightarrow v\)
            if \(v\) is not visited then
                \(v . p a t h \leftarrow u\).path followed by R
                Enqueue \(v\) to \(\mathbf{Q}\)
            end if
        end if
        if (Last move on u.path \(\neq R\) or u.path \(=\) null ) then
            Execute L on \(u \rightarrow v\)
            if \(v\) is not visited then
                \(v . p a t h \leftarrow u\).path followed by L
                Enqueue \(v\) to Q
            end if
        end if
    end while
```


## V. Results

Comparison of the number of moves required to sort $R_{n}$ with $L R E$ 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 $L R E, L R E 1$ and Search respectively.

## CONCLUSION

The first known upper bound for sorting $R_{n}$ with $L R E$ has been shown. A tighter upper bound has been derived. The

| $\boldsymbol{n}$ | 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 | 54 | 34 | 34 |
| 10 | 73 | 43 | 43 |
| 11 | 95 | 54 | 53 |

future work consists of identifying the exact upper bound for sorting $R_{n}$ with $L R E$. The identification of the diameter of the $L R E$ 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] Kuppili1 S. S., Chitturi1, B., and Srinath T., "An Upper Bound for Sorting $R_{n}$ with LE," ICACDS 2019.
[3] Kuppili1 S. S., Chitturi1, B., "Exact Upper Bound for Sorting $R_{n}$ 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 $O\left(n^{3}\right)$ 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 $S_{n}$ 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):205213, 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.


[^0]:    ${ }^{1}$ This article is submitted to 10 th International Advanced Computing Conference, IACC-2020.

