The traditional algorithm for sorting gives a bound of $O(n \log n)$ expected time without randomization and $O(n)$ with randomization. Recent researches have optimized lower bound for deterministic algorithms for integer sorting [1-3]. Andersson has given the idea of Exponential tree which can be used for sorting [4]. Andersson, Hagerup, Nilson and Raman have given an algorithm which sorts $n$ integers in $O(n \log \log n)$ expected time but uses $O(m^{\frac{n}{2}})$ space [4, 5].
Andersson has given improved algorithm which sort $n$ integers in $O(n \log \log n)$ expected time and linear space but uses randomization [2, 4]. Yijie Han has improved further to sort $n$ integers in $O(n \log \log n)$ expected time and linear space but passes integers in a batch i.e. all integers at a time [6]. These algorithms are very complex to implement. In this paper we discussed a way to implement the exponential tree sorting and later compare results with traditional sorting technique.

Reference

- Y. Han, M. Thorup, Sorting integers in $O(n \sqrt{\log \log n})$ expected time and linear space, IEEE Symposium on Foundations of Computer Science (FOCS’02), 2002.
- Y. Han, Deterministic sorting in $O(n \log \log n)$ time and linear space, 34th STOC, 2002.

Index Terms

Computer Science

Algorithms
### Key words

<table>
<thead>
<tr>
<th>Deterministic Algorithms</th>
<th>Sorting</th>
<th>Integer Sorting</th>
</tr>
</thead>
</table>

**Complexity**

- Space Requirement
- Exponential Tree