Generic Programming

Project

Team members:

1. K Shrinidhi Bhagavath, PES1201701525
2. Prince Jha, PES1201701608

Introduction

The chosen data structure is called LTree.Ltree grows like a tree and children are stored as a list, hence the name LTree (L -> list and Tree). This data structure is inspired by a data structure called VList which means vector of list. Our LTree is similar in some way but makes insertion faster.

Summary

As mentioned before LTree has a data part and stores children in the form of a list. Number of elements in children grows exponentially i.e., 1, 2, 4, 8, ... (child-1 has 1 element, child-2 has 2 elements, child-3 has 4 elements, child-4 has 8 elements, … for detailed view see figure in Abstract section). This data structure can be when we need fast random access along with fast insertion.

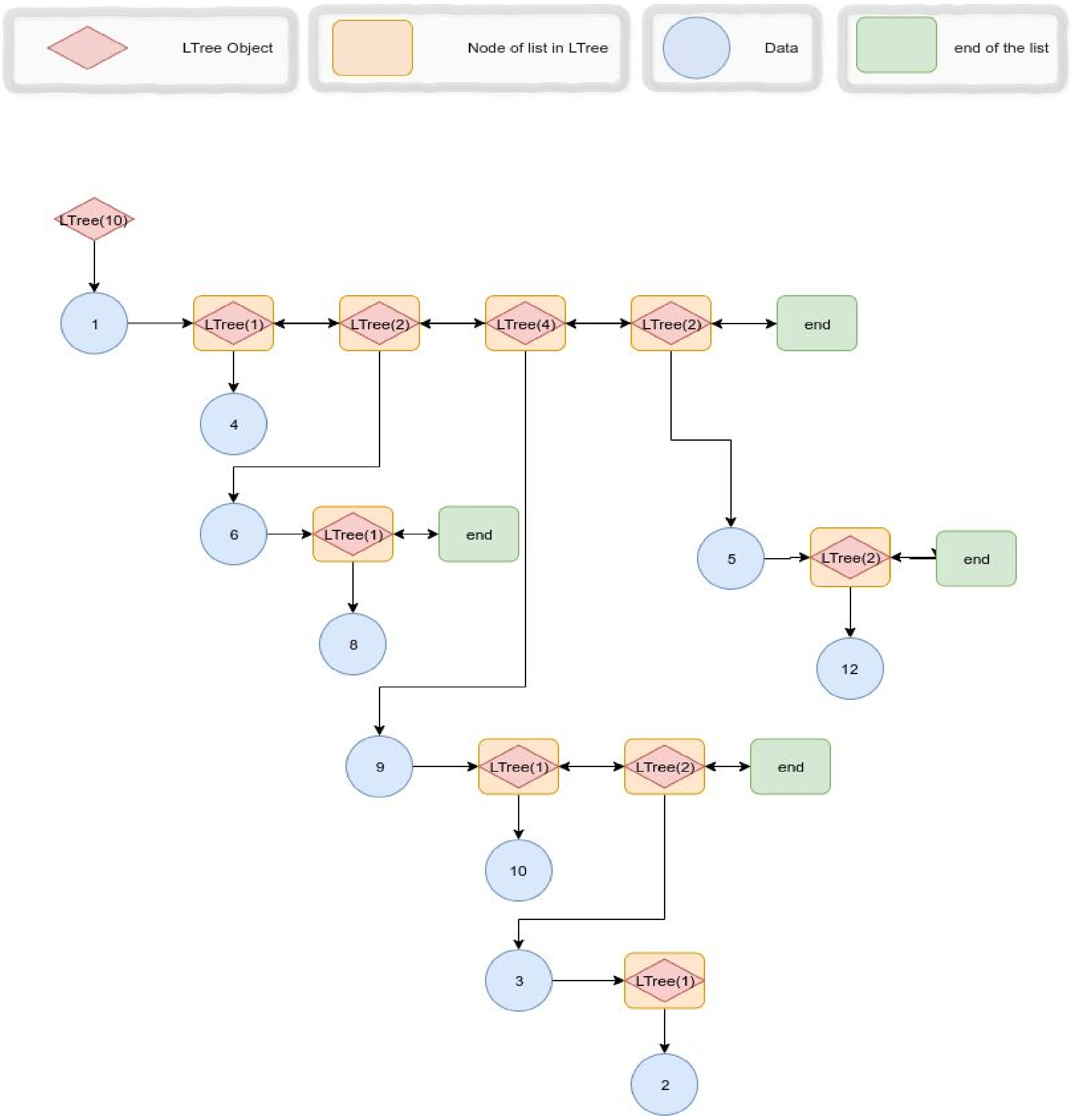
Let m = log(n)+log(n/2)+log(n/4)... until log gives 1.
for *2^64 = 18446744073709551616 elements* we get *m = 64+63+...+1 = 2080*

Iterator	:	Random Access Iterator
Operation on iterator ->		
++	:	O(1)
--	:	O(1)
-x	:	O(1) best case, O(m) worst_case
+x	:	O(1) best case, O(m) worst_case
-iterator	:	O(m)

Insertion at given position	:	O(m)
push_back	:	O(1) Best case,O(log(n)) worst case
pop	:	O(m)
indexing	:	O(m)

Ex

For data = {1, 4, 5, 6, 8, 9, 10, 3, 2, 5, 12}, LTree is in below figure


LTree Object
Node of list in LTree
Data
end of the list
LTree(10)
1
LTree(1)
LTree(2)
LTree(4)
LTree(2)
end
4
6
LTree(1)
end
5
LTree(2)
end
8
12
9
LTree(1)
LTree(2)
end
10
3
LTree(1)
2