# 19AIE203 - DATA STRUCTURES & ALGORITHMS 2

# S3 B. TECH CSE(AI)

# PROJECT REPORT

**Submitted by**


| Name | Roll Number |
|---|---|
| M DEVIKA | AM.EN. U4AIE20144 |
| NIVEDITA RAJESH | AM.EN. U4AIE20153 |



January 2022


Department of Computer science and Engineering
Amrita Vishwa Vidyapeetham
Amritapuri Campus
Kollam 690525
Kerala

# 19AIE203 - DATA STRUCTURES & ALGORITHMS 2

## ABSTRACT

To implement a segment tree and some of its applications like Lazy Propagation, Min and Max query in a range and persistent segment tree. Lazy propagation includes range related queries- the sum of given range & update nodes in each range. In this project, we are studying some theoretical aspects of the above-mentioned applications and implementing the same in java.

## INTRODUCTION

A segment tree is a data structure that is used to perform range queries and range updates. It is a height-balanced binary tree. It can solve range queries in O(log n) time. Each intermediate node of the tree represents a segment of the data set. They are useful when working with ranges of numerical data. In this project, the following use cases have been implemented.

1) Min and max query from a given range
2) Lazy propagation in segment tree
3) Persistent Segment tree.

The input elements are taken in the form of an array.

The min and max queries are to find the minimum and maximum from a specified range. The leaf nodes will be the elements of the input array while the internal node represents the minimum or the maximum respectively, of all leaves under it. The time complexity for tree construction is O(n). There is a total of 2n-1 nodes, and the value of every node is calculated only once in tree construction. The time complexity to query is O(log n). To query a range minimum and maximum, we process at most two nodes at every level and the number of levels is O(log n).

In lazy propagation, the sum of a specified range of elements, updating elements of a specified range and then taking the sum of that range of elements is done. The leaf nodes will be the elements of the input array while the internal node represents the sum of all leaves under it. A lazy array of the same size as the input array is taken and initialized to 0. A value 0 in the lazy array indicates that there are no pending updates on node ‘n’ in the segment tree. A non-zero value of a lazy array means that this amount needs to be added to node ‘n’ in the segment tree before making any query to the node. The time complexity is O(log N) where N is the range in which the value is being updated.

In the persistence segment tree, the segment tree can retain changes without losing initially assigned values. For each change in the segment tree, a new version is created. The initial

version is considered as version 0, and with each update, a new version is created. And for each update, at most log n nodes gets modified. The new version consists of only these nodes. For example, if for a given segment tree, node 'n' is modified, and it affects node 'a' and 'b', then new versions of node 'n', 'a' & 'b' are created and connected to the initial version. The time complexity will be the same as the query and point operation in the segment tree since we consider the extra node creation to be done in O(1). Hence, the overall time complexity per query for the new version and range sum query will be O(log n).

## PROBLEM AND SOLUTION

- **Lazy propagation:**

1) Start with the root of the segment tree.
2) If the array index to be updated is not in the current node's range, then return
3) Else update current node and recur for children.

Input: A array of elements to form a segment tree and the range in which the sum of the values must be calculated and the range in which the value is to be updated along with the value to be updated.

Output: Original sum and the updated sum of values in each range

- **Persistent segment tree**

For a given segment tree, if a node is updated, then all the affected nodes are taken, and a new version is created with only these nodes. Values are then assigned to the new nodes. Similarly, all the versions of the segment tree are created. To keep track of the versions, keep track of their root node. Then for each range sum query, we will pass the required version's root node in our query function and output the required sum.

Input: An array and different point update operations.

Output: The sum of elements in range r1 to r2 just after the nth update.

- **Min and max query from a given range**

**Minimum:**

The leaf nodes will be the elements of the input array while the internal node represents the minimum of all leaves under it. If the range of node is within the start query index and end query index, then the value in the node is returned. Else, if the range of node is completely outside started query index and end query index, then "UNABLE TO GET VALUE" is printed,

else, the method is called with respect to the right and left child. (Range with respect to array index).

Input: A array of elements to form a segment tree and the range in which the minimum query should be calculated

Output: Minimum of values in the given range is printed.

**Maximum:**

The leaf nodes will be the elements of the input array while the internal node represents the maximum of all leaves under it. If the range of node is within the start query index and end query index, then the value in the node is returned. Else, if the range of node is completely outside started query index and end query index, then “UNABLE TO GET VALUE” is printed, else, the method is called with respect to the right and left child. (Range with respect to array index).

Input: A array of elements to form a segment tree and the range in which the minimum query should be calculated

Output: Maximum of values in the given range is printed.

## REFERENCES:

https://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/
https://www.geeksforgeeks.org/segment-tree-set-1-range-minimum-query/
https://www.geeksforgeeks.org/segment-tree-set-2-range-maximum-query-node-update/
https://www.geeksforgeeks.org/persistent-segment-tree-set-1-introduction/
https://www.geeksforgeeks.org/lazy-propagation-in-segment-tree/
https://www.hackerrank.com/topics/segment-tree
https://www.educative.io/edpresso/what-is-a-segment-tree
https://medium.com/nybles/understanding-range-queries-and-updates-segment-tree-lazy-propagation-and-mos-algorithm-d2cd2f6586d8
https://www.baeldung.com/cs/segment-trees

## CONTRIBUTION

- Nivedita Rajesh: Implementation of Persistent segment tree and Min & max query from a given range
- M Devika: Implementation of Lazy Propagation (update segment tree and sum in the given range)