Segment Trees in Algorithmic Problems

Piotr Szczepaniak

March 30, 2025

Contents

Abstract

This document provides an overview of segment trees. In the first place I will describe some algebraic topics which are necessary for better understanding how and why segment trees works. This knowledge will be useful for reading the rest of the paper where We will dive into different kinds of trees. For each structure, I will explain how it work and how to apply it to problems. Then, I will look at each structure's time complexity and space complexity.

1 Foundations of Segment Trees

A segment tree is a binary tree used for storing information about segments. To efficently retrieve or update informations about elements stored in segment tree we can perform various operations, the most common of which is the range query, range update or point update (which is sipler case of range update). One of the examples of use can be maximum value of elements in given range or sum of elements in given range.

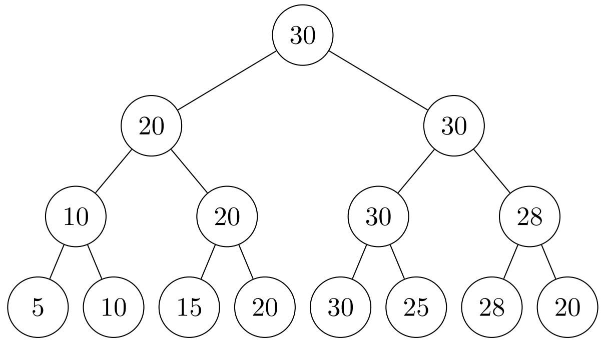

Figure 1: Example of a segment tree with maximum value of elements in given range.

1.1 Operation types

To ilustrate use case of segment tree we will construct tree with max value on segment. Let's say we are given an array $A = [5, 10, 15, 20, 30, 25, 28, 20]$ of length $n = 8$. For now let's assume that the input array is of size $n = 2^k$ where k is integer (for different sizes of input we will fill input array with neutral elements (see section 2) to make it's length a power of 2). The height of tree is $h = \log 2n$. Let's define $dep(i)$ as depth of node i in our tree. We can see that $dep(root) = 0$ and $dep(leaf) = h$. We want to be able to perform the following operations:

- **Build structure**
  We will create a segment tree from an array. To build a segment tree need to create a binary tree where each node will store the maximum value of elements in its subtree.

Algorithm 1 Build Segment Tree for Maximum on Segment (Iterative)

---

procedure BUILDTREE(arr, seg)
    for $i = 0$ **to** $n - 1$ **do** ▷ Fill leaves of the segment tree
        $seg[n + i] \leftarrow A[i]$
    end for
    for $i = n - 1$ **downto** 1 **do** ▷ Calculates nodes from bottom to top
        $seg[i] \leftarrow \max(seg[2 \times i], seg[2 \times i + 1])$
    end for
end procedure

This approach of building a segment tree gives us an array seg where $seg[n, n * 2 - 1]$ are values form input and value at

- **Point update**
  Change the value of an element at a given index.
- **Range query**
  Find the sum of elements in a given range.

1.2 Monoids

A monoid $(S, e, *)$ is a set equipped with an associative binary operation $S \times S \to S$ and an identity element e.

- **Associativity**
  For all $a, b, c \in S$, $(a * b) * c = a * (b * c)$.
- **Identity element**
  There exists an element $e \in S$ such that for all $a \in S$, $a * e = e * a = a$.