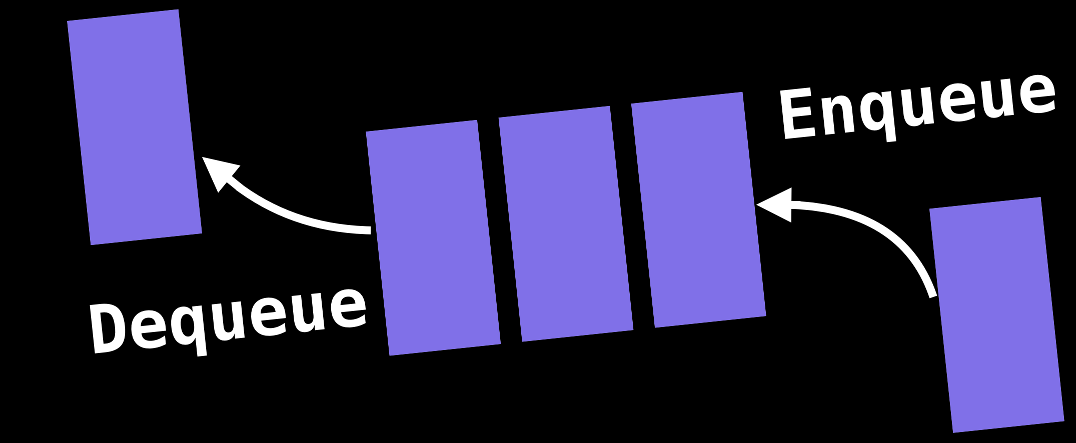
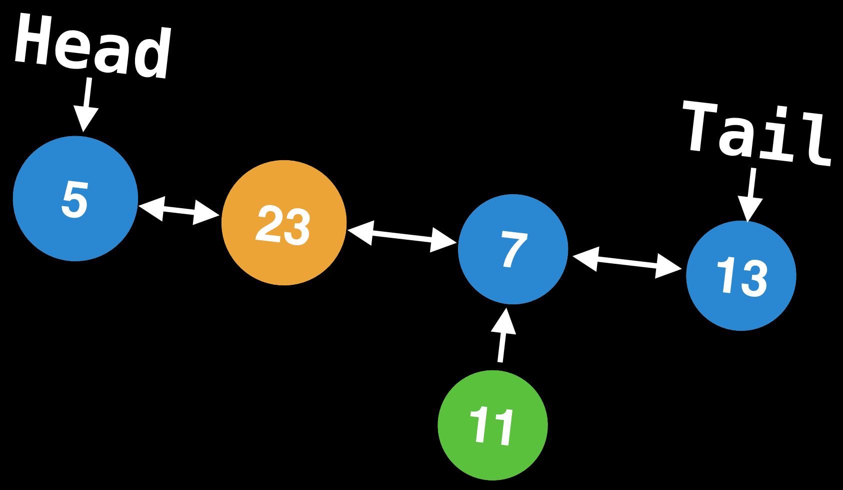
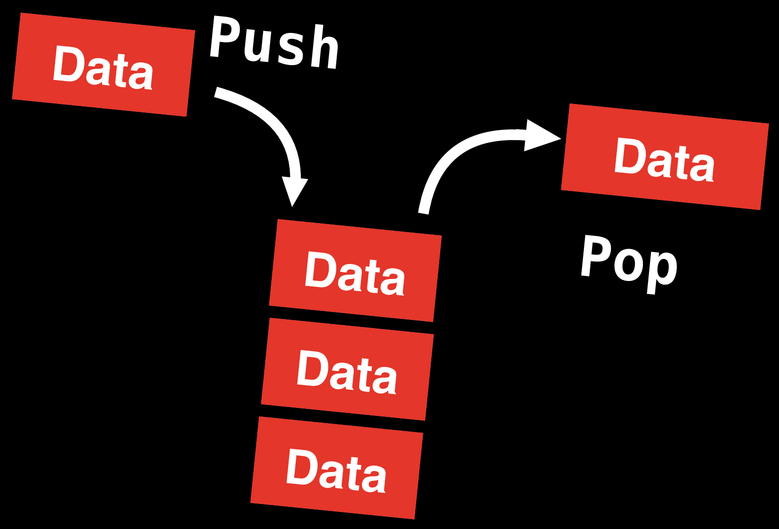
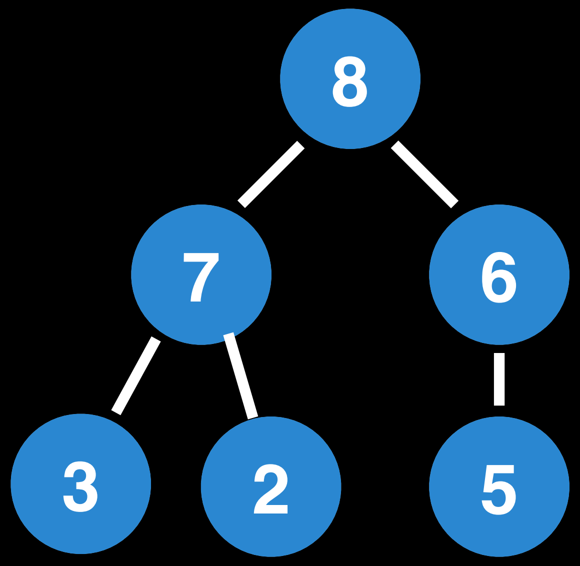
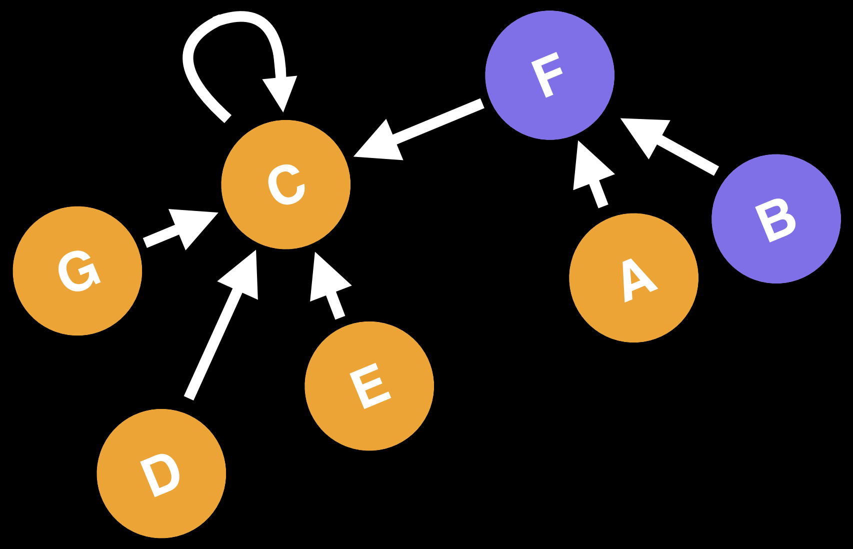




Data Structure Video Series







Segment Trees

A data structure for dynamic range updates
and range queries

William Fiset

Motivation

Given a **static array** of input values, we want to be able to do **range queries** and **range updates** on the input array.

Many data structures are able to do range queries on a static array (e.g Fenwick trees & Sparse tables), but the ability to easily do **range updates** is what distinguishes the segment tree from the others.

In terms of time complexity, a standard segment tree can be built from the input array in **O(n)** with linear space, and answer range queries in **O(log(n))**, as well as perform range updates in **O(log(n))** (using lazy propagation).

Motivation

Given a **static array** of input values, we want to be able to do **range queries** and **range updates** on the input array.

Suppose we start with the following list of numbers:

2	1	3	8	3	-3	5	6	7	2	0	5	-1

Motivation

Given a **static array** of input values, we want to be able to do **range queries** and **range updates** on the input array.

Suppose we start with the following list of numbers:

2	1	3	8	3	-3	5	6	7	2	0	5	-1

An example of a **range update** might be: add a value of \`6\` to all the elements in the range [4, 7]:

2	1	3	8	3	-3	5	6	7	2	0	5	-1
				+6	+6	+6	+6					

=

2	1	3	8	9	3	11	12	7	2	0	5	-1

Motivation

Given a **static array** of input values, we want to be able to do **range queries** and **range updates** on the input array.

The updated array of values:

2	1	3	8	9	3	11	12	7	2	0	5	−1

After the range update, we might be interested in querying the sum of all the elements in the range [2, 6]:

2	1	3	8	9	3	11	12	7	2	0	5	−1

$$3 + 8 + 9 + 3 + 11 = 34$$

Range query/update matrix

	Assign range update	Addition range update
Min range query		
Max range query		
Sum range query		

Range query/update matrix

	Assign range update	Addition range update	Multiplication range update
Min range query	No issues	No issues	Supporting negative numbers may be difficult[2]
Max range query	No issues	No issues	Supporting negative numbers may be difficult[2]
Sum range query	No issues	No issues	No issues
GCD range query	No issues	Impossible to do range updates?[1]	No issues

[1] There doesn’t appear to be a way to support gcd range queries with addition updates (AFAIK).

[2] Supporting negative multiplication updates for max/min range queries may be more difficult/require tracking additional information (see https://github.com/williamfiset/Algorithms/issues/208)

Header

motivation

static nature of the input data, growing the segment tree is not really feasible…?

types of query/update functions (sum update, max query combo)

Storage and representation

Building the segment tree

queries

point updates + range query

range updates (lazy propagation) + point query

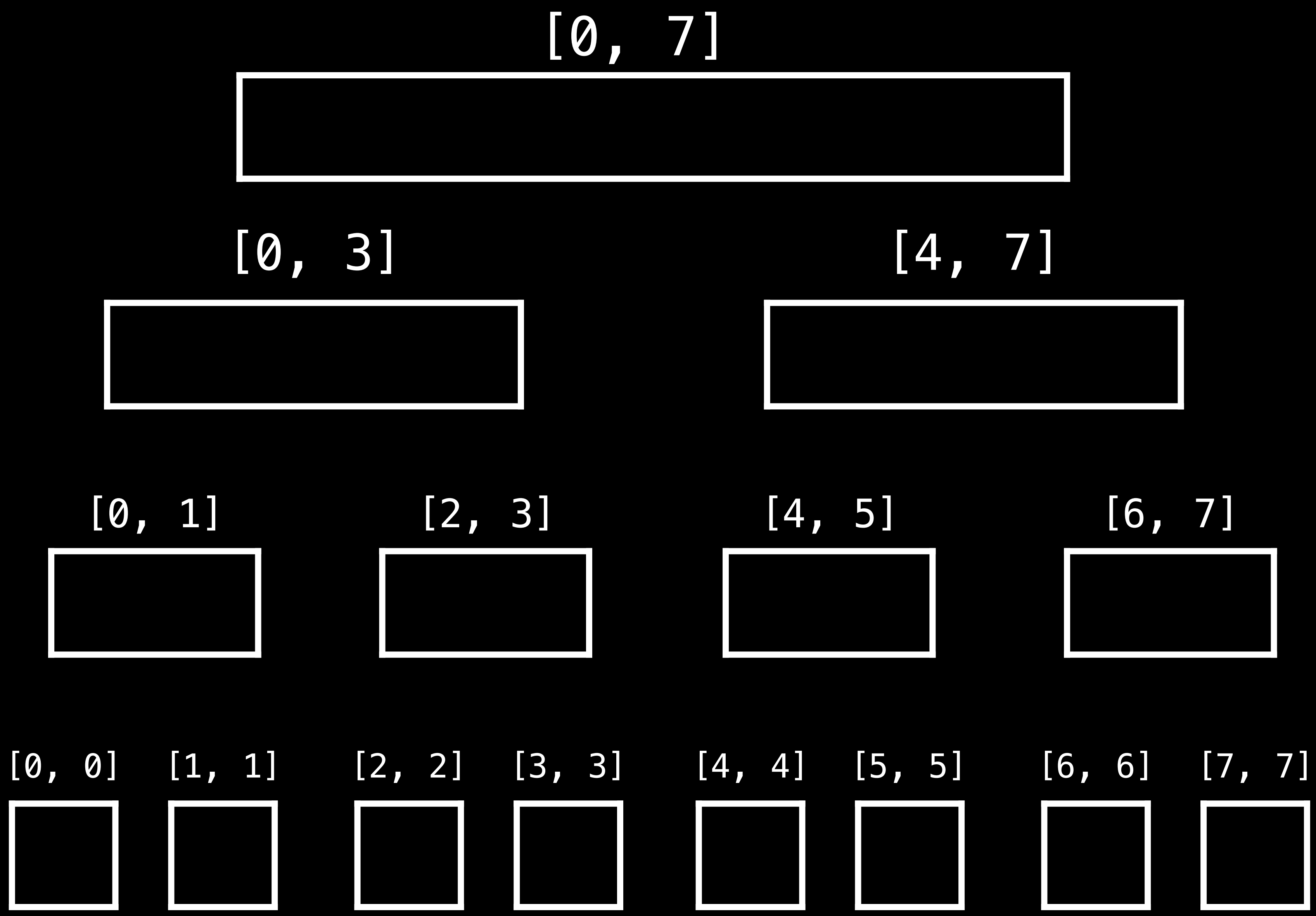
range update + range query

coordinate compression

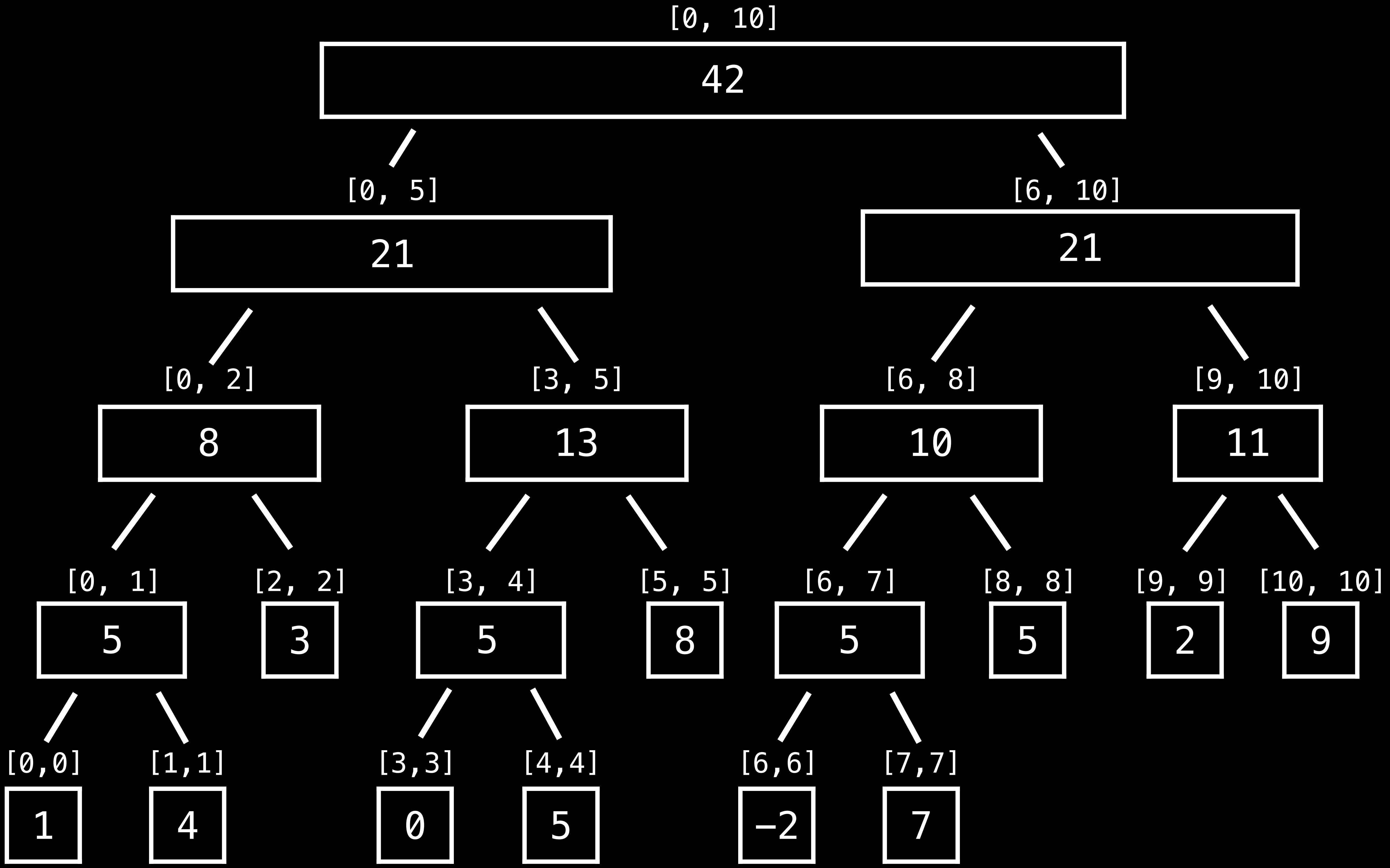
todo(waf): test you segment tree via Kattis

Building a segment tree from an input array is usually done bottom up. Fill the leaves with the input values and combine segments recursively up the tree using the **segment combination function**.

1	4	3	0	5	8	-2	7



0	1	2	3	4	5	6	7	8	9	10
1	4	3	0	5	8	−2	7	5	2	9



Next Video: <insert video name here>

Source Code Link

Implementation source code can be found at the following link:

github.com/williamfiset/algorithms

Link in the description:

References

https://cp-algorithms.com/data_structures/segment_tree.html#toc-tgt-4