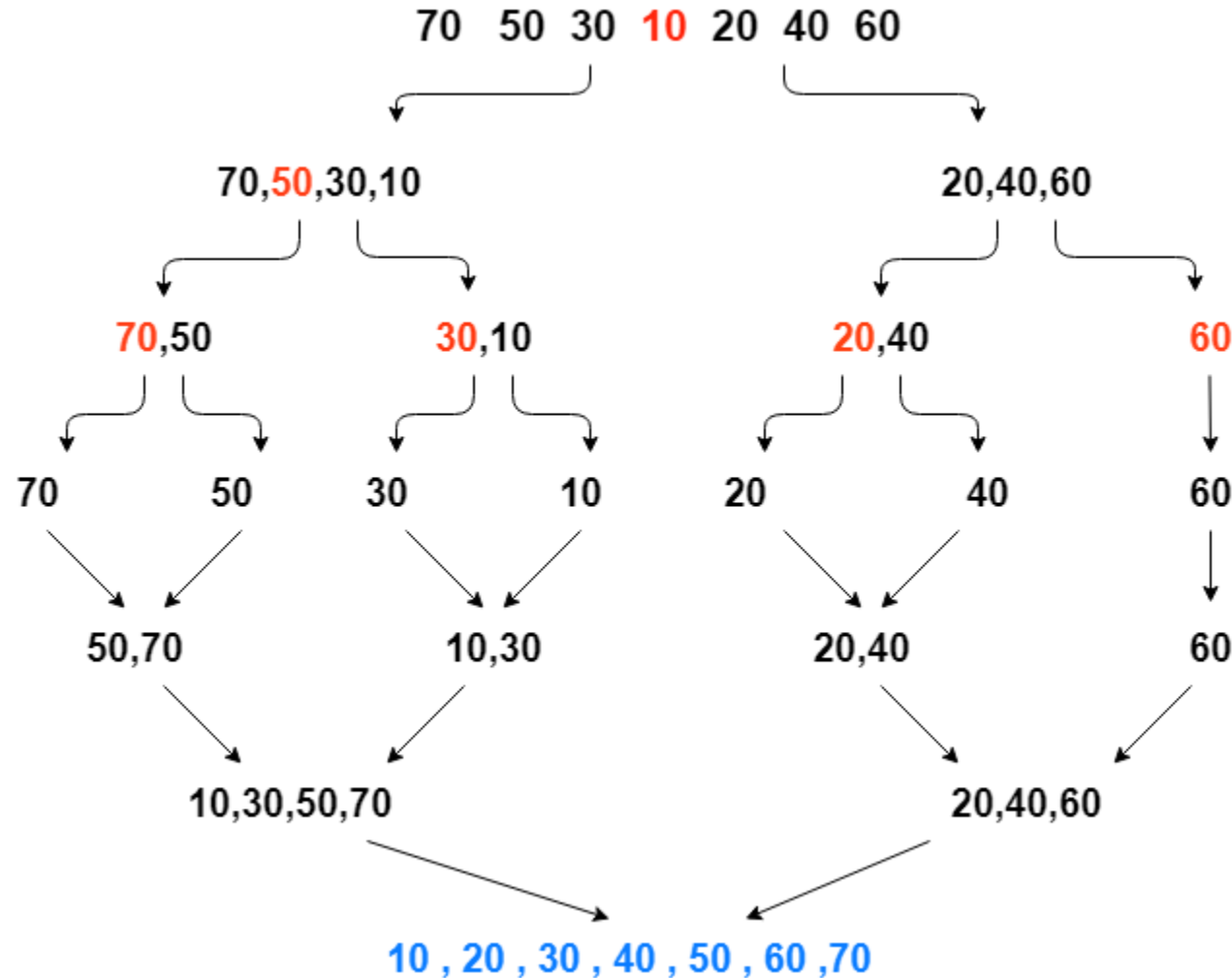


# *COMP-352*

## *Tutorial #5*

---

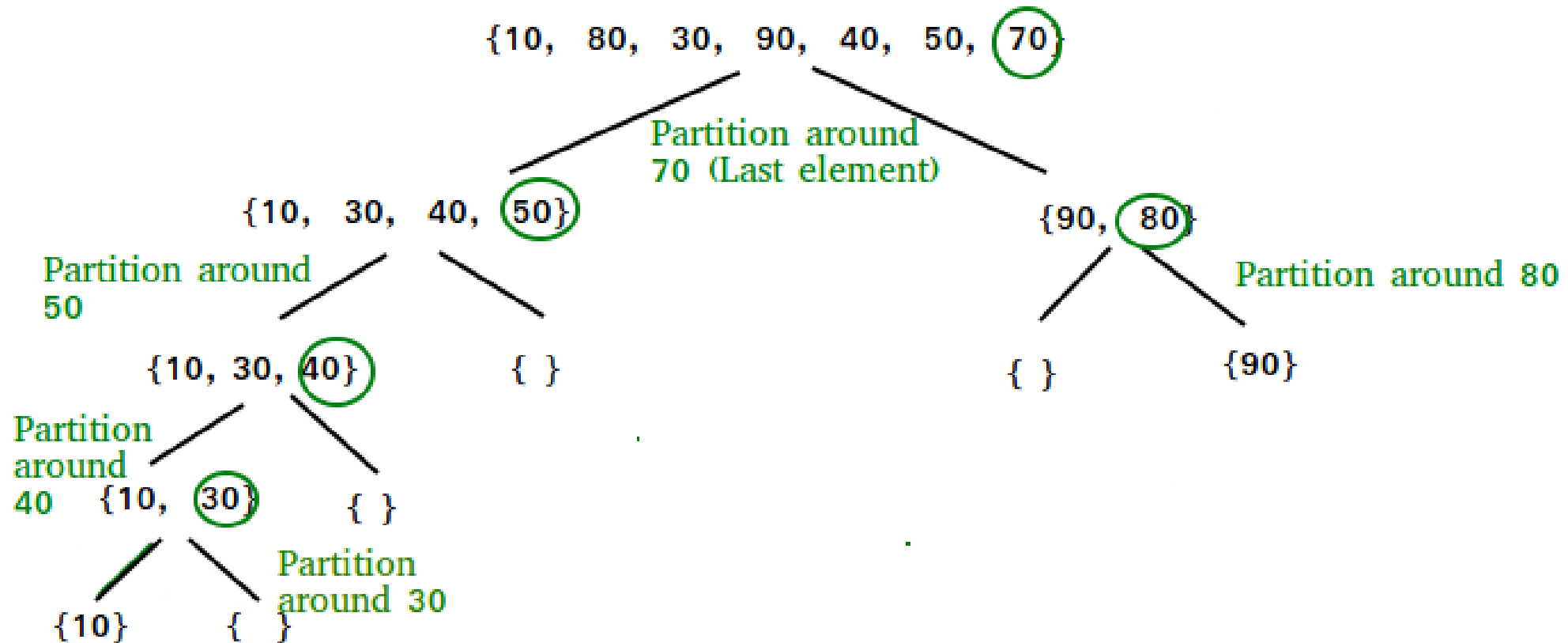
# CASE: MergeSort (Look at the code)



- Unsorted Array
- Split into 2
- Split again
- Split for last time
- Merge based on order
- Merge again
- Final merge

Source: <https://www.digitalocean.com/community/tutorials/merge-sort-algorithm-java-c-python>

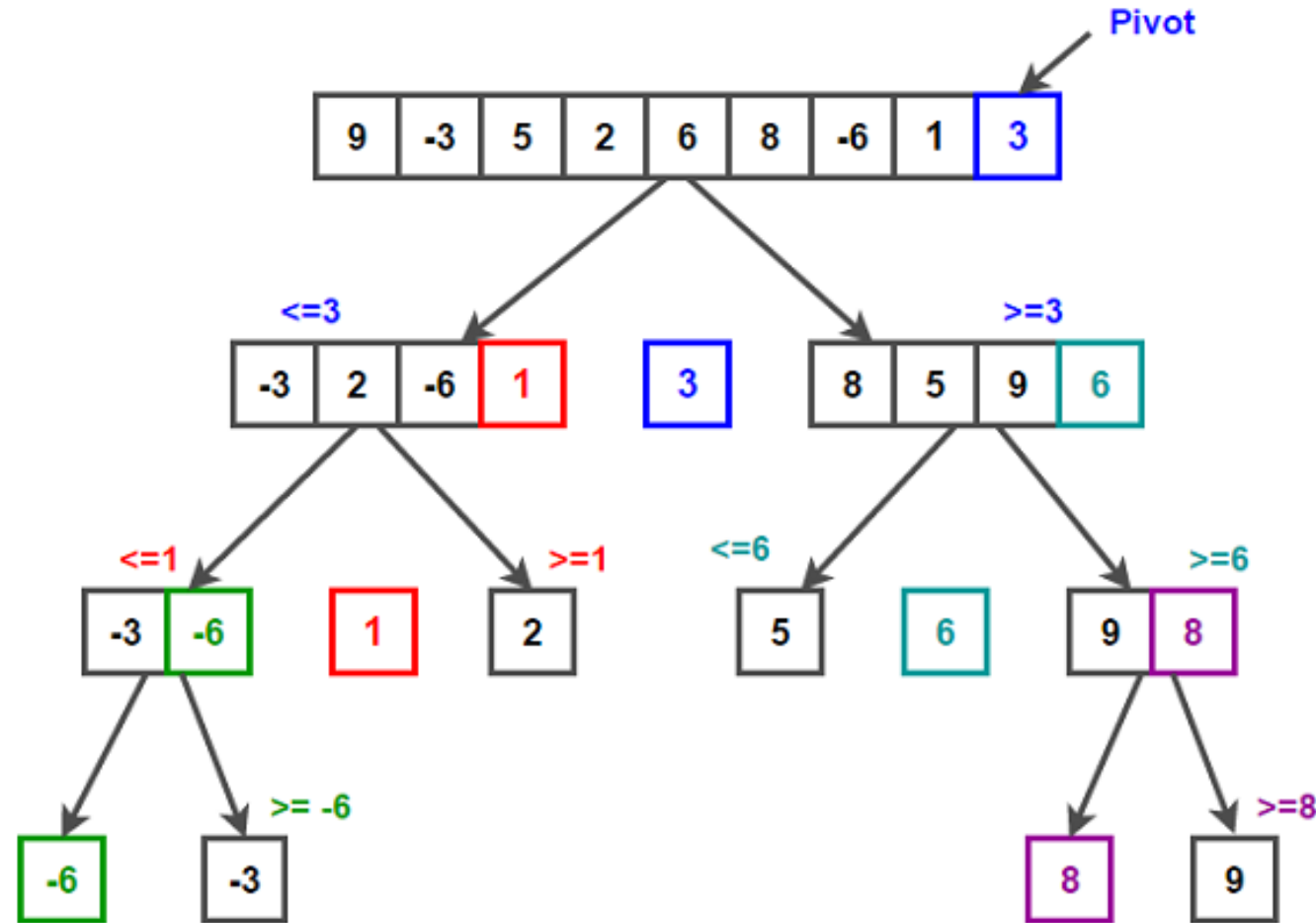
# QuickSort: MergeSort's Sibling



-> **QUESTION:** Which methodology is used in Quick Sort? How is it achieved?

Source: <https://www.geeksforgeeks.org/quick-sort/>

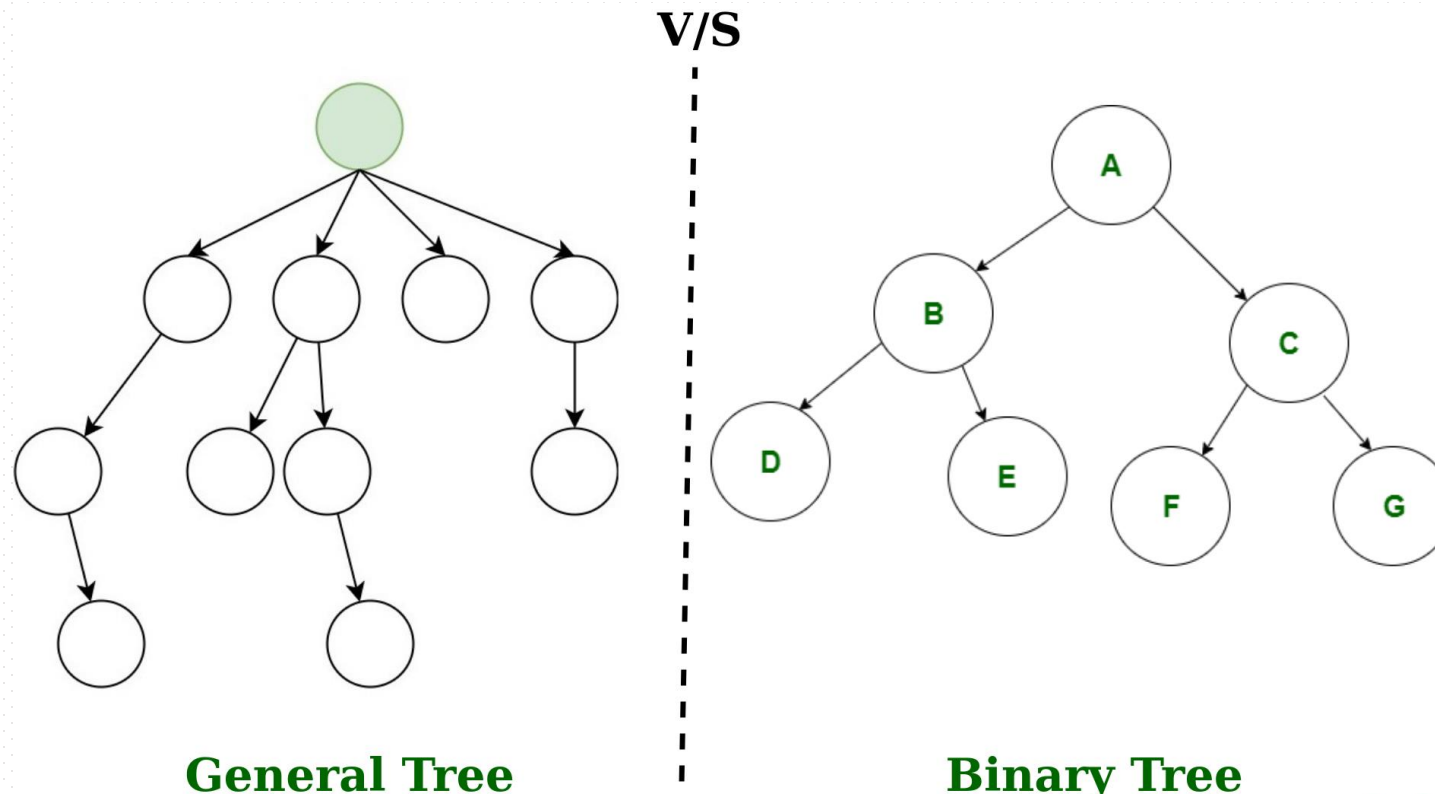
# QuickSort: An alternative approach



Source: <https://www.techiedelight.com/quicksort/>

# DS: Trees and Binary Tree

- **Tree** (General) data structure is a **hierarchical (non-linear)** data structure. It is a collection of nodes that are connected by edges and has a hierarchical relationship between the nodes.
- **Binary Tree:** Each node can have at most 2 child nodes.



# Heap Structure

---

- A specialized complete binary tree where the parent is either smaller or bigger than its children.

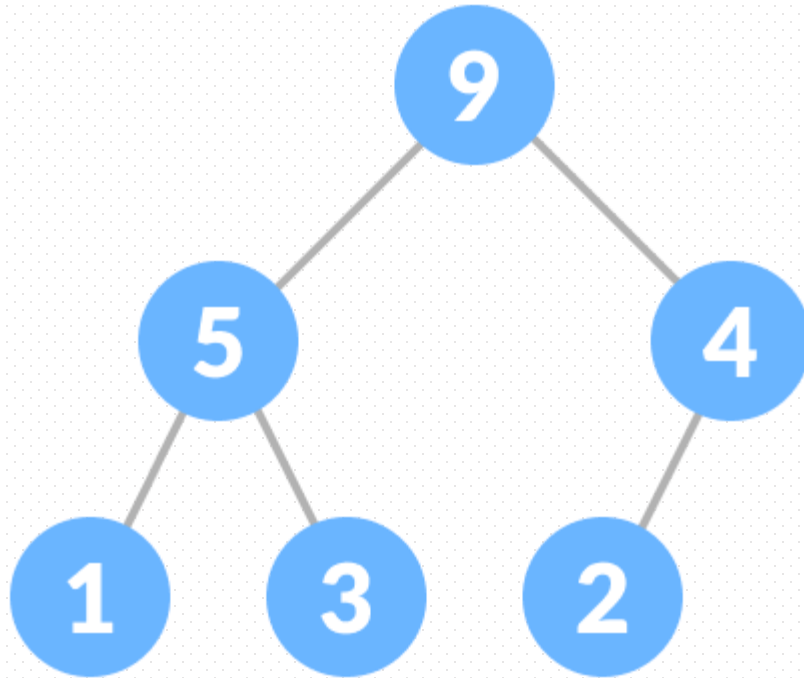
## **It satisfies the heap property:**

- Any given node is always greater than its child node/s and the key of the root node is the largest among all other nodes. This property is also called **max heap property**.
- Any given node is always smaller than the child node/s and the key of the root node is the smallest among all other nodes. This property is also called **min heap property**.

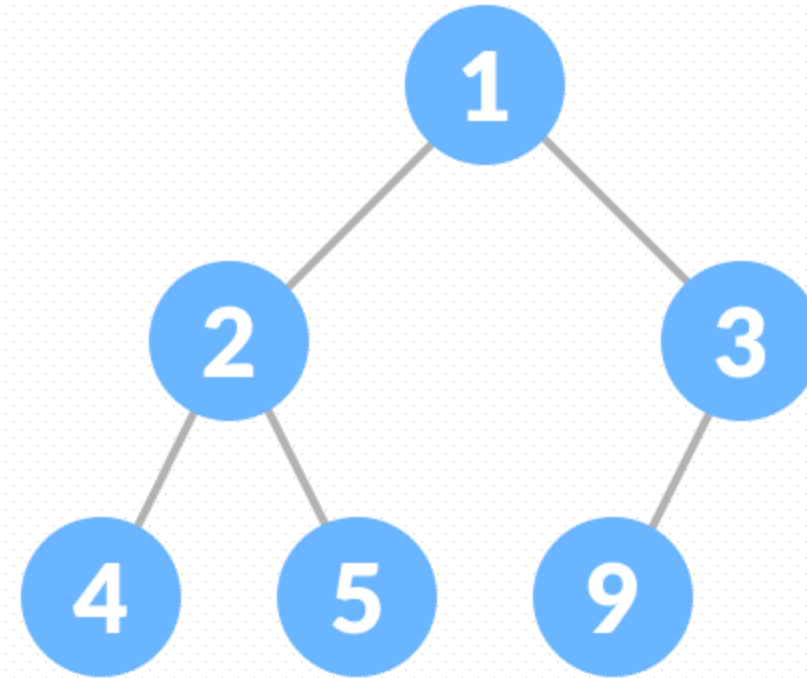
# The Two Heaps

---

**MAX HEAP**

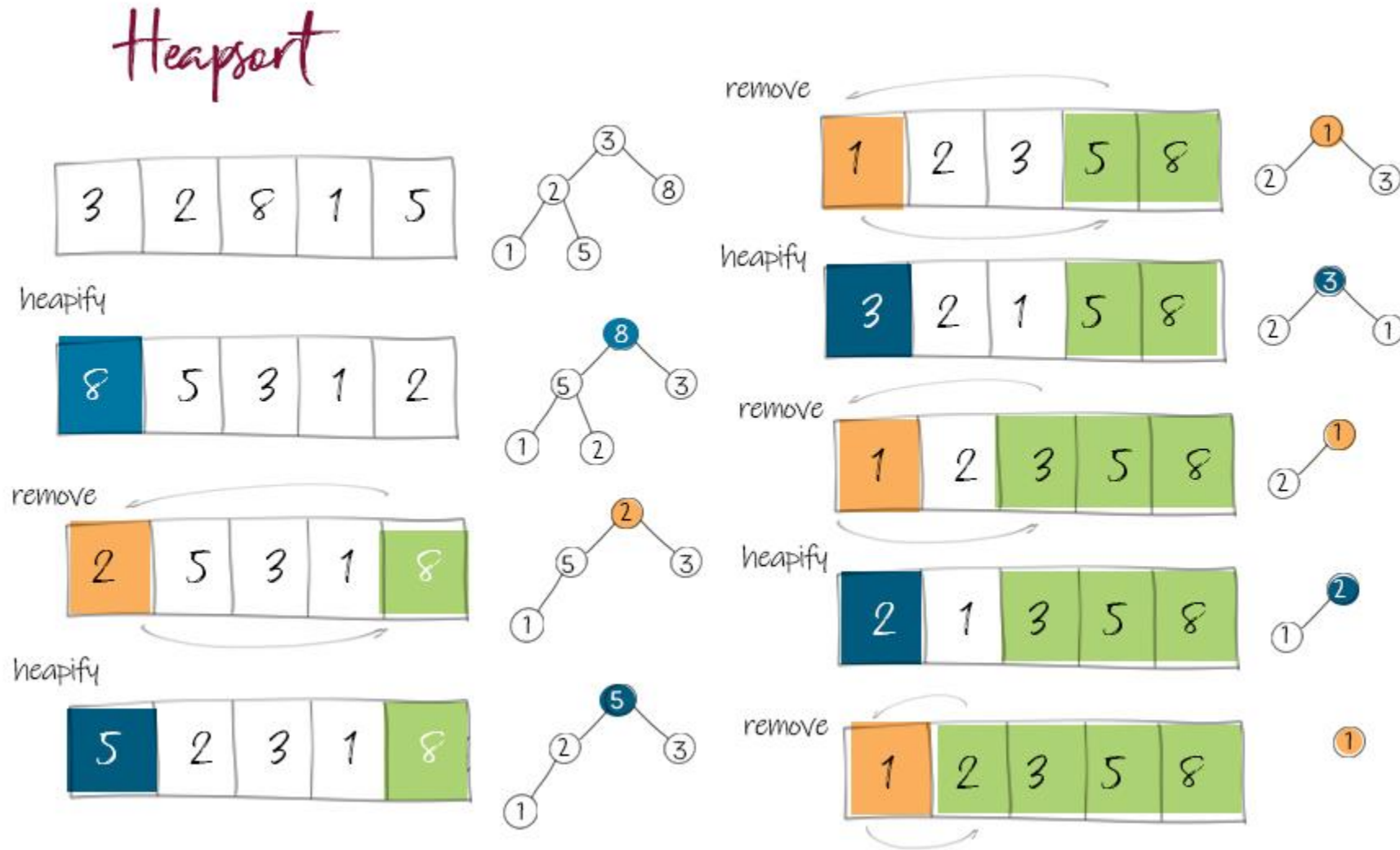


**MIN HEAP**



Source: <https://www.programiz.com/dsa/heap-data-structure>

# HeapSort: A Different Story



Source: <https://www.lavivienpost.net/heapsort-iteration-and-recursion/>



# Algorithms Comparison

Sorting Algorithms	Time Complexity			Space Complexity
	Best Case	Average Case	Worst Case	Worst Case
Bubble Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Quick Sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(n)$
Heap Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$

Source: <https://www.enjoyalgorithms.com/blog/comparison-of-sorting-algorithms>

# Exercise: Better Quicksort

---

**Task:** Implement a median of three pivot selection algorithm for the quickSort algorithm.

- **Median of three:** selecting the median between the first, middle, and last elements in the array.
- Use the quickSort implementation given on the GitHub repo or use your own if you have one.
- Does this make the quickSort algorithm better?

*THANK YOU*

---