COMP-352 Tutorial #4

ALGORITHMS

Definition: A set of rules to be followed in calculations or other problem-solving operations.

- ❖ Basically, anything involving processes and instructions.
  - Searching, Sorting, creation of data structures, accessing…etc.

- AS SIMPLE AS adding two numbers.
- AS COMPLEX AS human neural network communications.

Searching

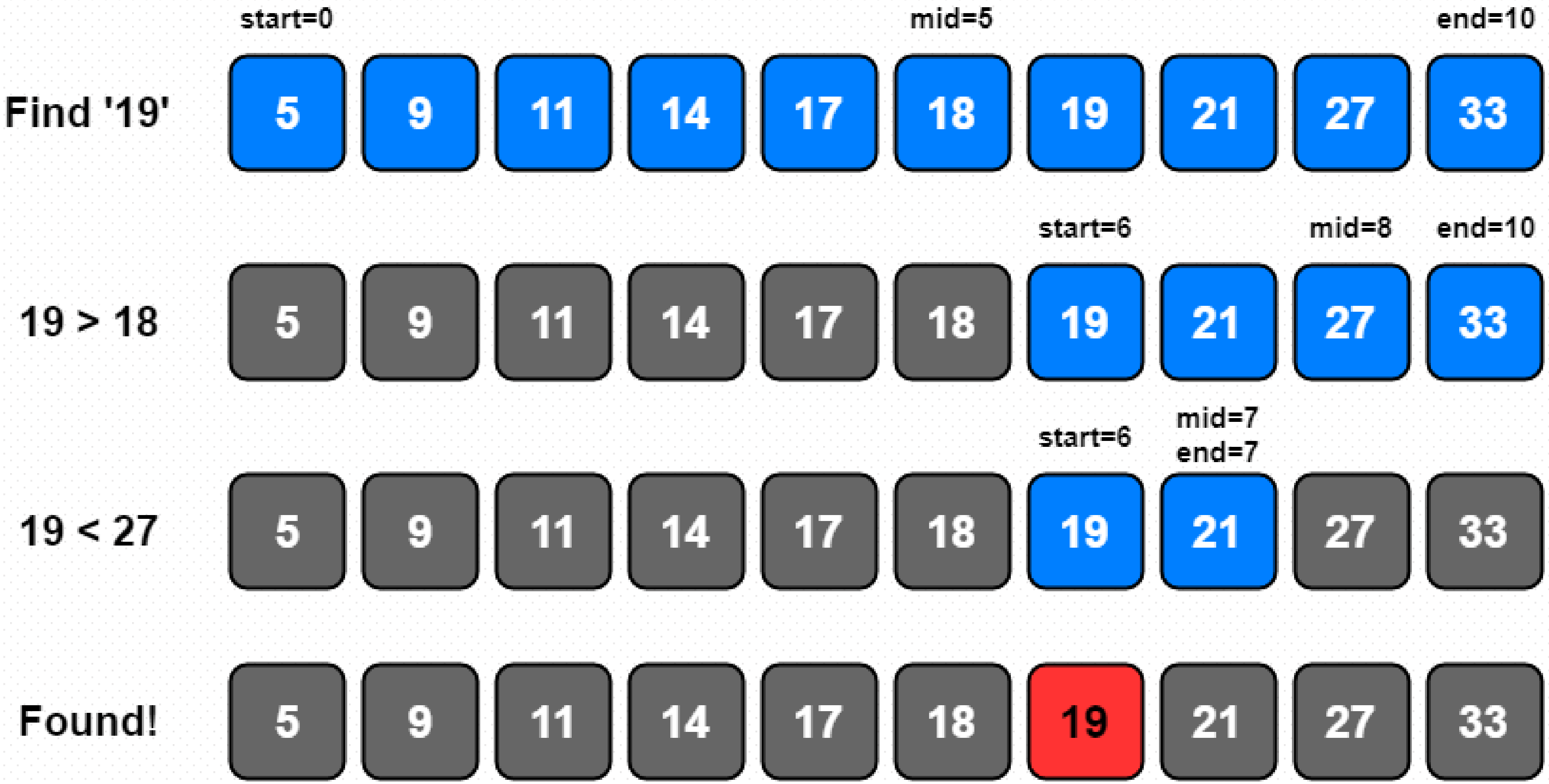
Definition: Algorithms designed to check for an element or retrieve an element from any data structure where it is stored.

Two types:

- Sequential Search
- Interval Search
- Interpolation Search

-> QUESTION: What is the most important component in searching algorithms? What is the performance figure?

CASE: Binary Search



-> QUESTION: Which methodology is used in Binary Search?

Source: https://emre.me/coding-patterns/modified-binary-search/

Sorting

Definition: An algorithm used to rearrange a given list of elements in a desired order.

- Sorted data is always useful since we can be more systematic with our algorithms, i.e., they make more sense.

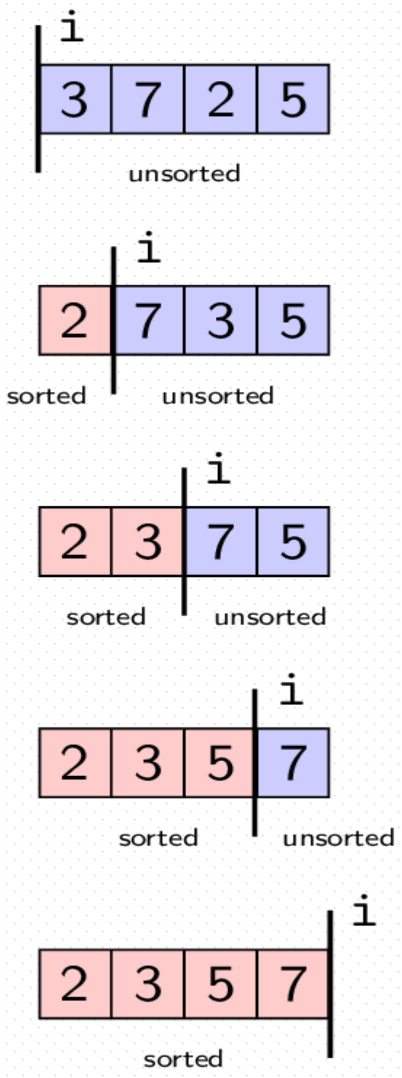
TYPES: *Why linear and non-linear?*

- Linear sorting
- Non-linear sorting
- *Non-comparative sorting*

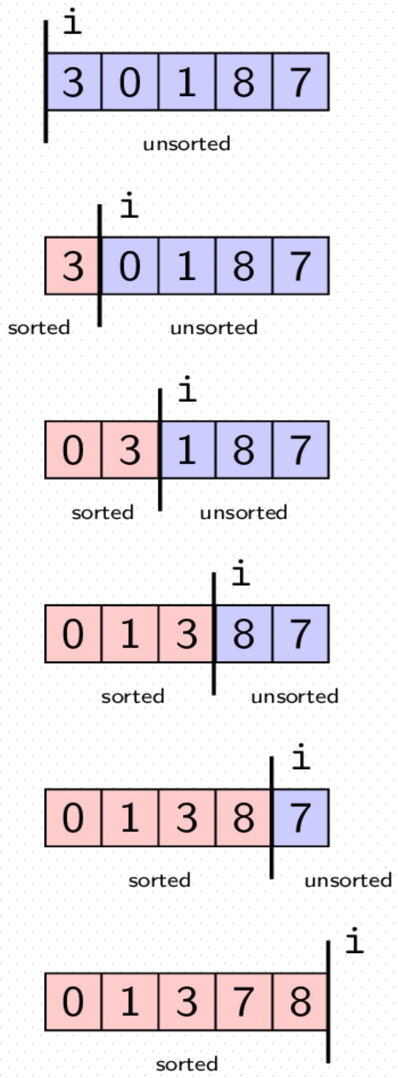
-> QUESTION: What are the most important components in Sorting algorithms? Why are data structures important in sorting?

CASE: The 3 Linear Sorts

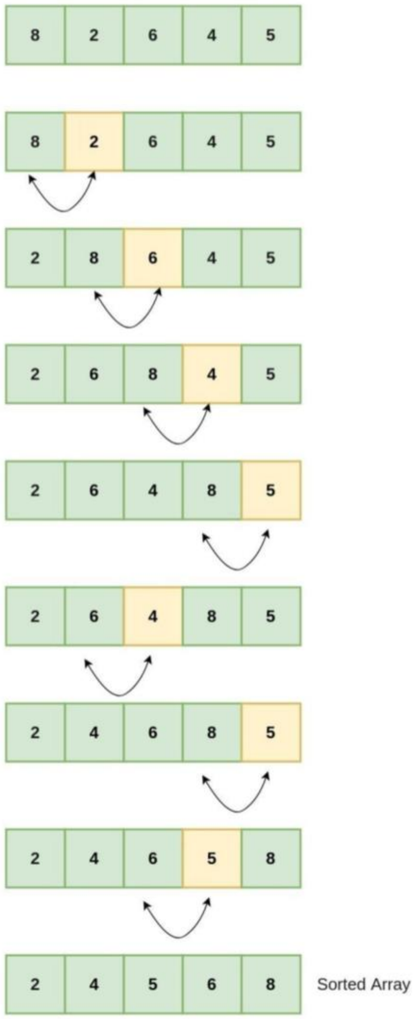
Selection Sort



Insertion Sort



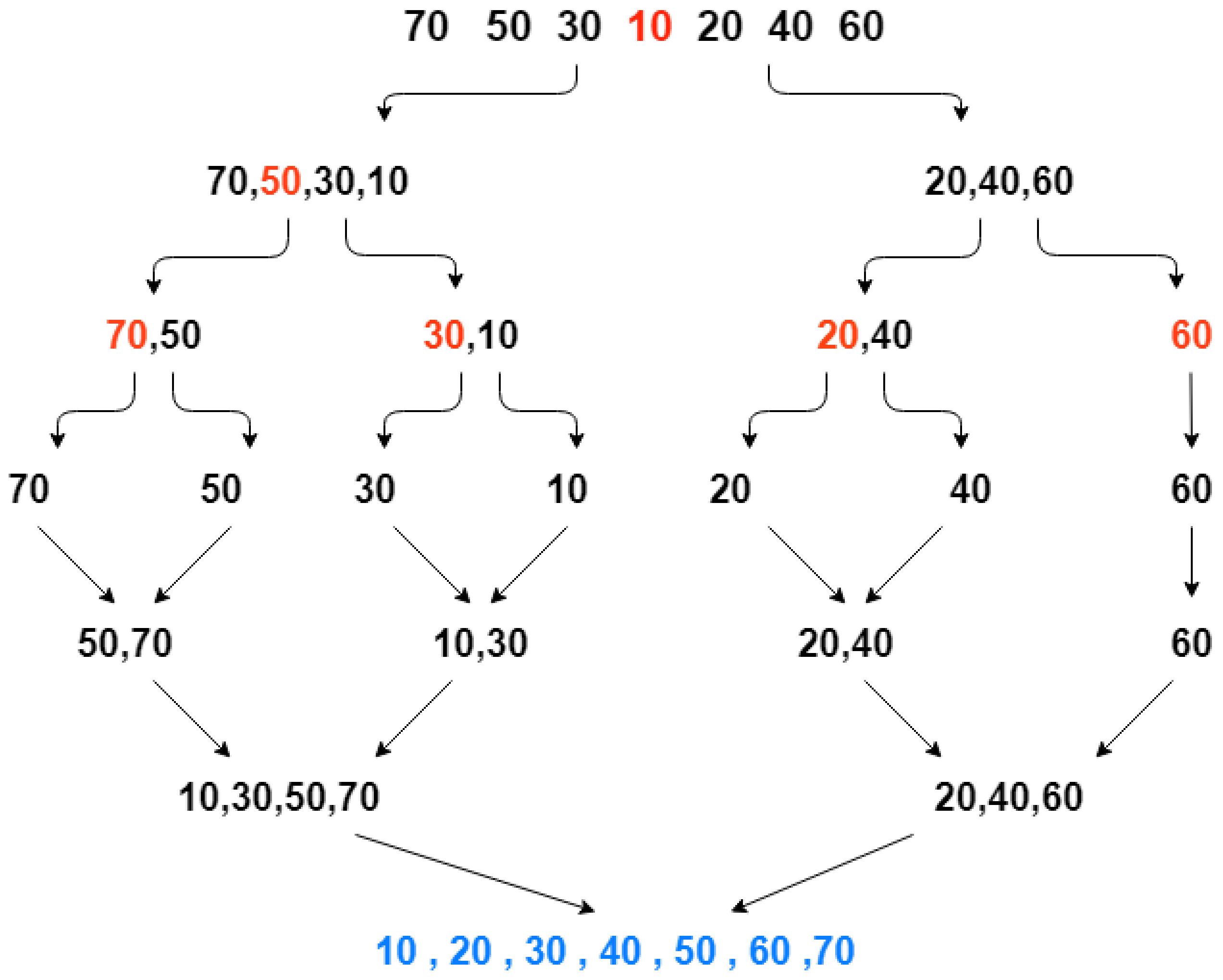
Bubble Sort



Sources:

- https://www.cs.toronto.edu/~david/course-notes/csc110-111/
- https://codeforgeek.com/bubble-sort-algorithm-in-python/

CASE: MergeSort (non-linear sorting)



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

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

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

Why so many sorts: Algorithm’s Analysis

What do Time & Space signify: Time & Space Complexity?

- Efficiency
- Speed
- Constraints

Quantifying our algorithms for comparison: What is n & asymptotic analysis?

- **Big O:** It defines the upper bound and **upper bound** on an algorithm is the most amount of time required.
- **Big Ω:** It defines the lower bound and **lower bound** on an algorithm is the least amount of time required.
- **Big Θ:** It defines the tightest bound and **tightest bound** on an algorithm is a sandwich between upper and lower bound.

Classes of Measure: Best, Average, and Worst cases.

* *Best, Average, Worst ≠ Big Ω , Big Θ, Big O*

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(nlogn)	O(nlogn)	O(nlogn)	O(n)

-> **QUESTION:** Which notation is used?

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

Exercise: Enhancing Linear Sorting algorithm

TASK #1: Optimized Bubble Sort implementation:

- Using the given implementation of BubbleSort.java in the COMP-352 repo, make it optimized.
- **Hint:** *Use a Boolean value.*

TASK #2: Adding Compares and Swaps counting functionality.

- Pick one of the linear sorting algorithms from the COMP-352 repo and add the counter variables ‘compare’ and ‘swap’
- Increment these values where appropriate and output their values.
- These variables can be kept public for convenience.

THANK YOU