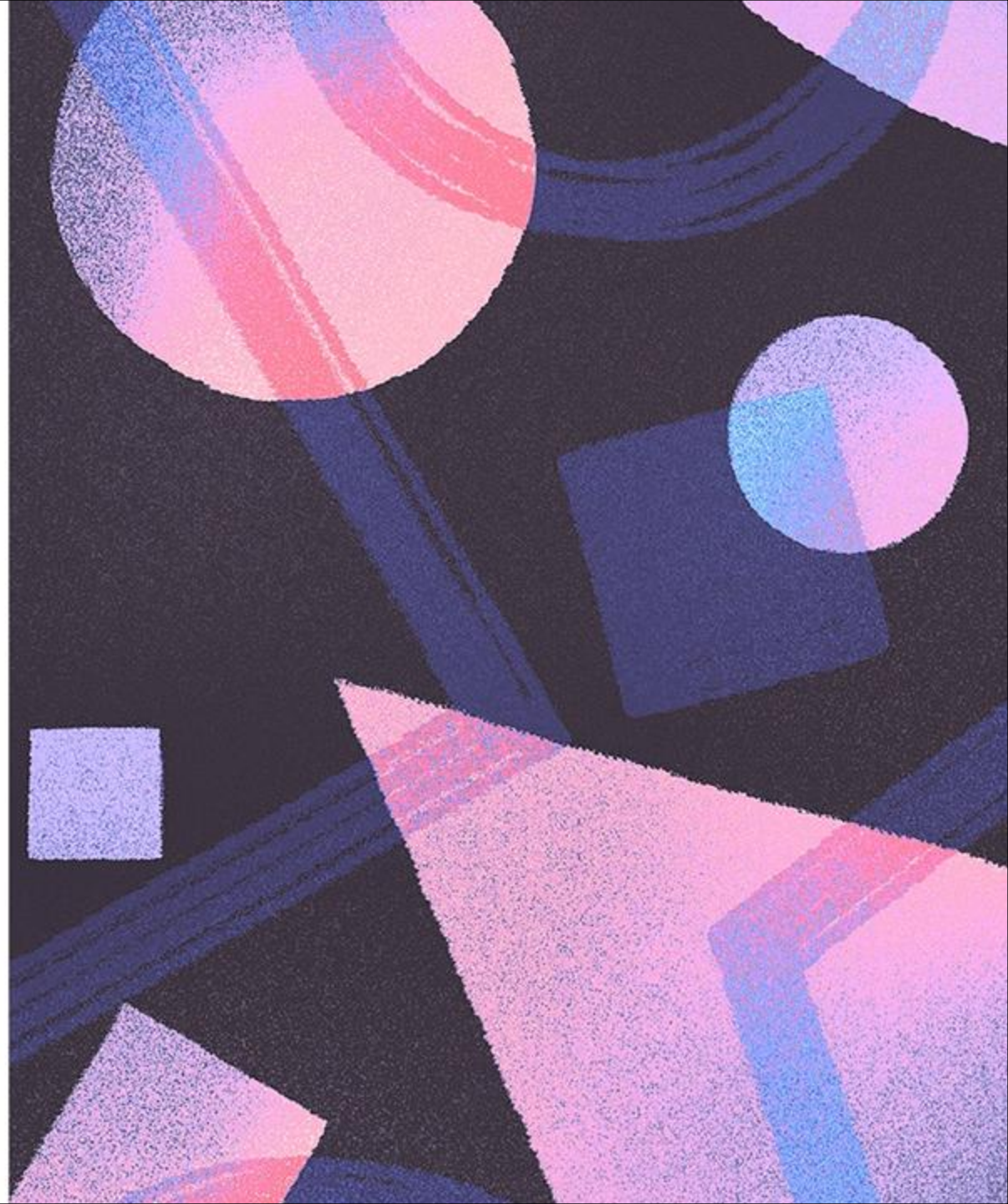

Data Structures Lab
Lecture 8

> Sorting & Selection Algorithms

Maryam Skaik, Jan-2026

01: Why Sorting Is Important?

- Searching becomes faster
- Data becomes structured
- Many algorithms assume sorted input
- Real-life examples: databases, rankings, scheduling
- Sorting is often a preprocessing step

Classification of Sorting Algorithms

- In-place vs not in-place
- Stable vs unstable

02: Bubble Sort

- Repeatedly swap adjacent elements if out of order
- Largest elements “bubble” to the end
- In-place, Stable
- Complexity: $O(n^2)$ time, $O(1)$ space

03: Selection Sort

- Find minimum element
- Swap with first unsorted position
- In-place, Unstable
- Complexity: $O(n^2)$ time, $O(1)$ space

04: Insertion Sort

- Insert each element into its correct position
- Like sorting cards in hand
- In-place, Stable
- Complexity: Best O(n), Worst O(n^2), Space O(1)

05: Merge Sort

Idea

- Divide array into halves
- Sort each half recursively
- Merge sorted halves

- Not in-place, Stable
- Complexity: O(n log n) time, O(n) space

06: Quick Sort

Idea

- Choose pivot
- Partition array
- Recursively sort partitions

- In-place, Unstable
- Complexity: O(n log n), Worst $O(n^2)$, Space O(log n)

THANK YOU