Advanced Data Structres

Advanced Data Structures & What They Store

Data Structure	What It Stores	Notes
array	stores elements in a specific order	in Python, same as lists
queue	stores elements in a specific order	first in first out (FIFO)
stack	stores elements in a specific order	last in first out (LIFO)
tree	stores elements hierarchically	each node can have multiple children
binary tree	stores elements hierarchically	each node has at most 2 children
heap	stores elements by priority	always get smallest/largest first

Sorting Algorithms

Time Complexity (Big O)

What is it? A measure describing how runtime scales with input size

Why do we need it? To compare algorithm efficiency and predict performance on large inputs

How do I compute it? Count operations: constant = O(1), loops = O(n), nested loops = $O(n^2)$

Sorting Algorithm Examples

Bubble Sort (slow)

```
function bubble_sort(array):
    n = length of array

    for i from 0 to n-1:
        for j from 0 to n-2:
            if array[j] > array[j+1]:
                swap array[j] and array[j+1]

    return array
```

Merge Sort (fast)

```
function merge_sort(array):
    if length of array <= 1:
        return array

    mid = length of array / 2
    left = merge_sort(array[0 to mid])
    right = merge_sort(array[mid to end])

    return merge(left, right)


function merge(left, right):
    result = empty array
    i = 0, j = 0

    while i < length of left AND j < length of right:
        if left[i] < right[j]:
            append left[i] to result
            i = i + 1
        else:
            append right[j] to result
            j = j + 1

    append remaining elements from left to result
    append remaining elements from right to result

    return result
```