· Measures has runtime scales with input size

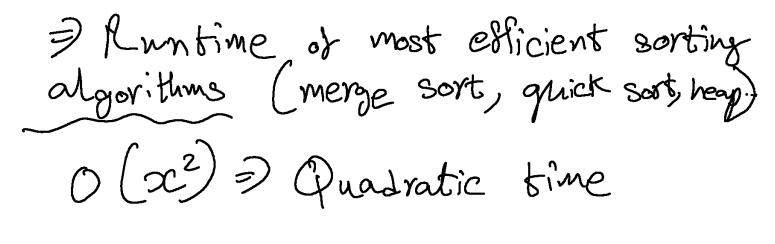
O(1) =) Constant time

Accessing an array by index
 Access a back table
 Binary search on sorted array
 Performing operations on a balanced binary tree

O(n) = Linear time

Runfine grows with input or each element is touched once "Finding min, max elements in an unsorted array · Checking it an element exists in an unsorted array

O(mlogn) => Linearithmic time



·Bubble Sort

· Nested Loops

O(n³) => Cubic time Naive matrix multiplication
Triple nested Loops O(2") = Exponential time · Recursive algorithms O(n) = Factorial time · Permutation - generation problems b Very inefficient for trivial input size

Important to note that big-O is not everything I its important to consider how abjorithmus work in the memory/rest. of the Array Access system Although accessing a 2D array is O(n2), accessivy by rows is faster than accessing by collumns! Reding row-suise maximises sequential access I More cake friendly Linked list us Array · Both are linear time · Accessing on array is faster BArray elements are closer together DLinked-list nodes are often scattered in memory