Sorting & Big-O notation

Applikationsudvikling - CS101

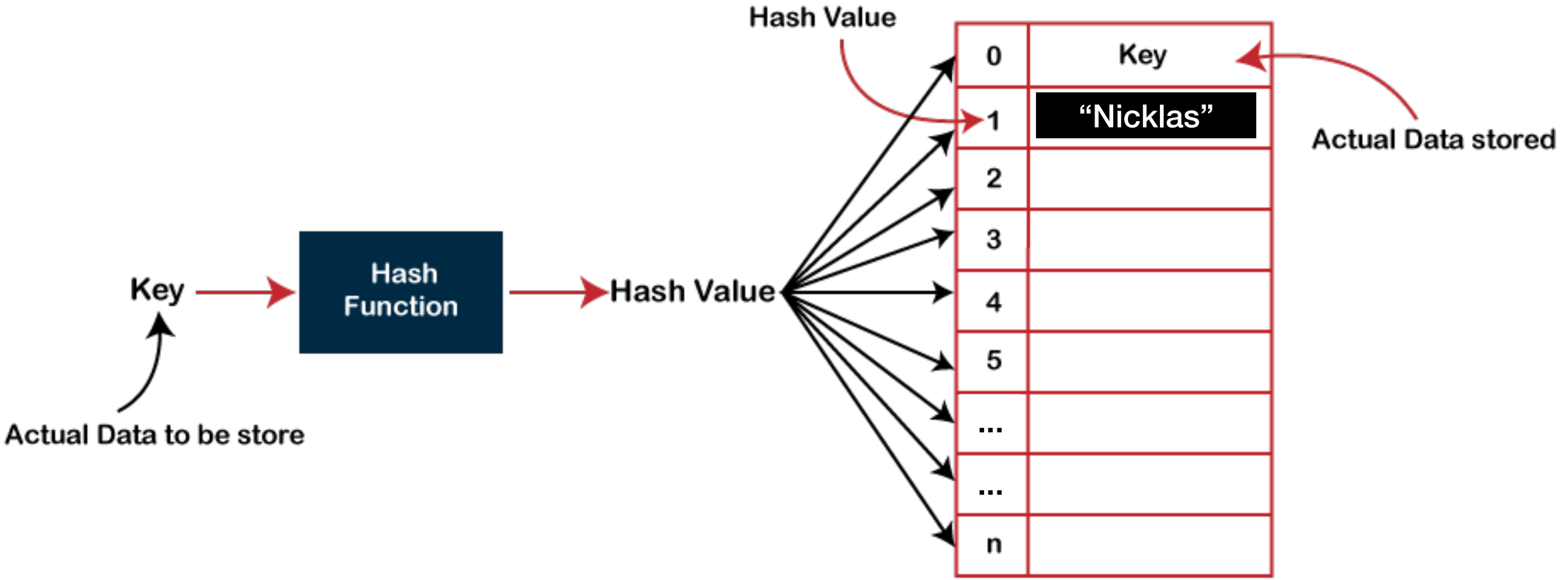
NIFR 2024

Agenda

Sorting & Big-O notation

- Time complexity Big-O notation
- Trading space for time “leetcode” for twosum
- Sorting algorithms
  - Bubble sort algorithm Pair Analysis
  - Insertion sort algorithm Pair programming
  - Selection sort algorithm Pair programming

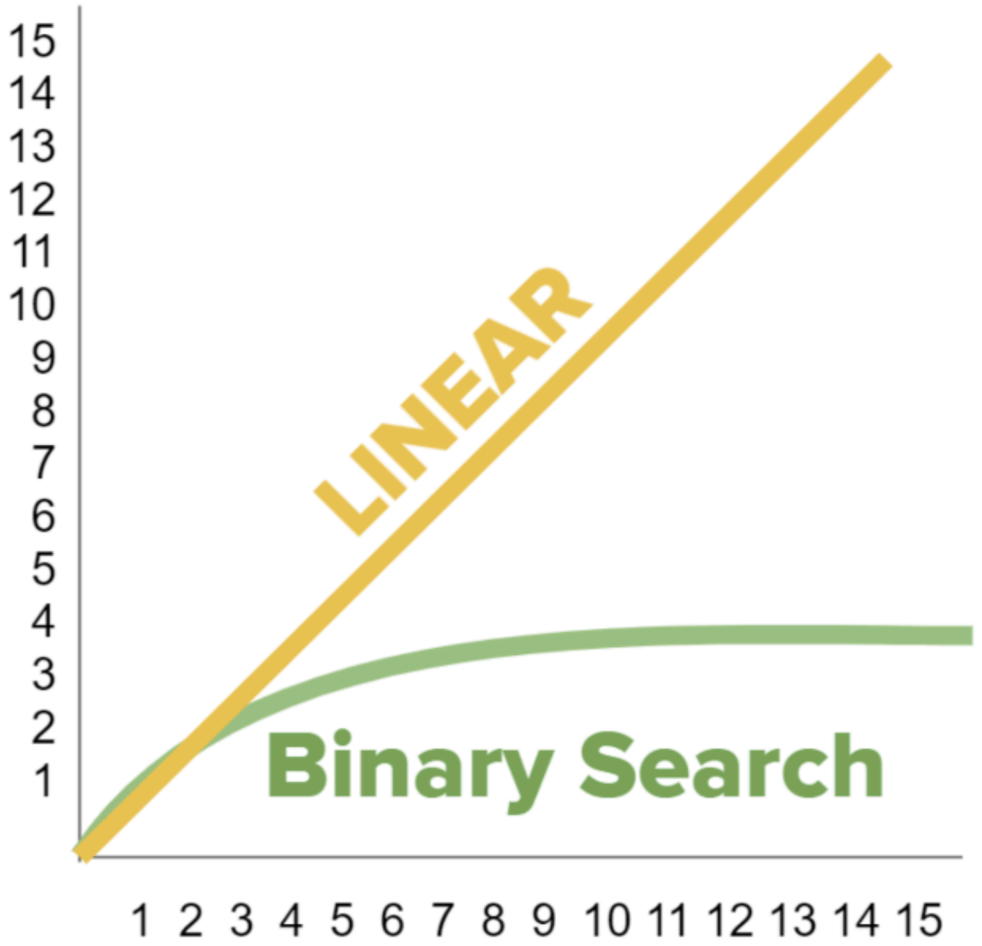
Hash Value	Key	Value
1	Nicklas	20436262
3	Karsten	20202020
12	Evander	29392291
	…	…
	N	N



What is meant by fast?

Order of growth

Time complexity



Order of growth

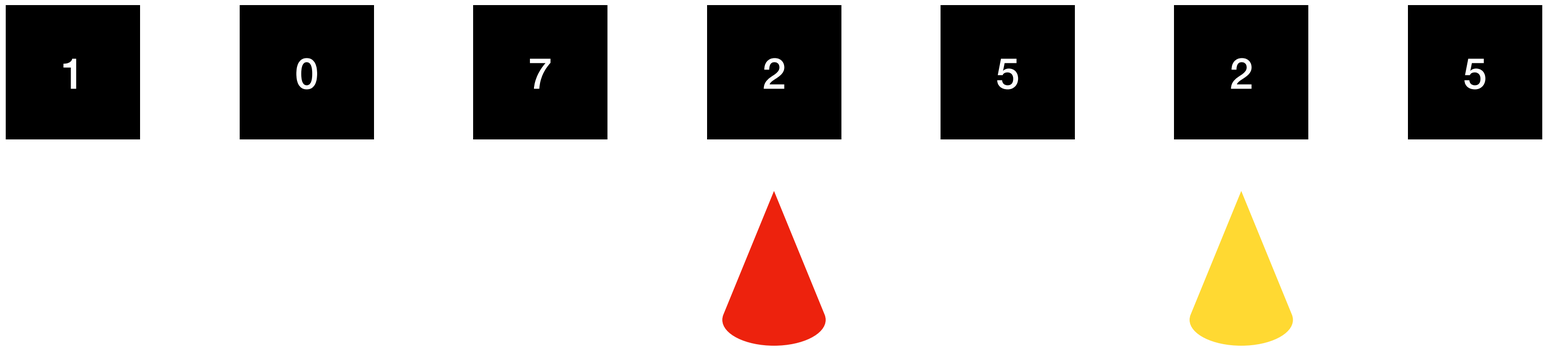
TwoSum function

```
fun twoSum(nums: IntArray, target: Int): IntArray {
    for (i in nums.indices) {
        for (j in i + 1 until nums.size) {
            if (nums[i] + nums[j] == target) {
                return intArrayOf(i, j)
            }
        }
    }
    throw IllegalArgumentException("No two sum solution")
}
```

Order of growth

TwoSum function

Target: 4



Order of growth

TwoSum function

Target: 4

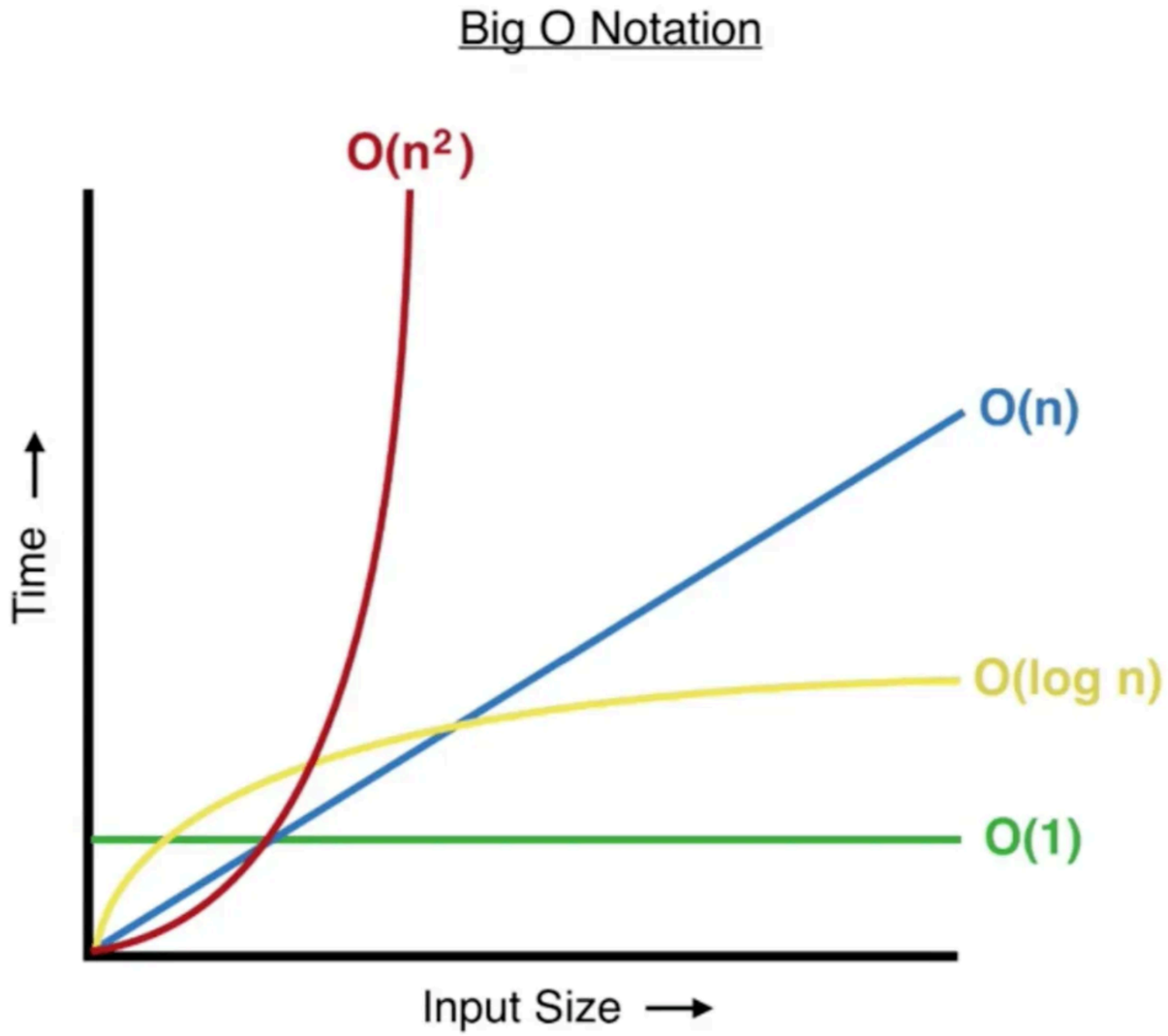
1	0	7	0	5	0	6	2	2

Not only 3 increase but 3*3 = 3^2 - 2

1	0	7	0	5	0	6	2	2

Not only 3 increase but 3*3 = 3^2 - 2

Solution has order of growth
O(n^2)



How does input influence execution time?

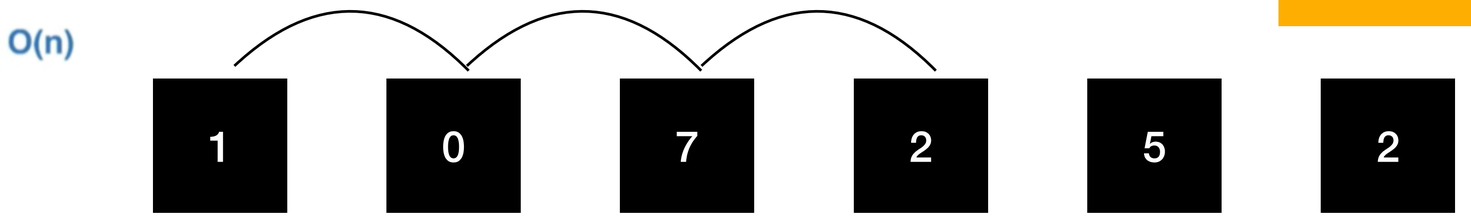
TwoSum function - observe the loops

$O(n^2)$

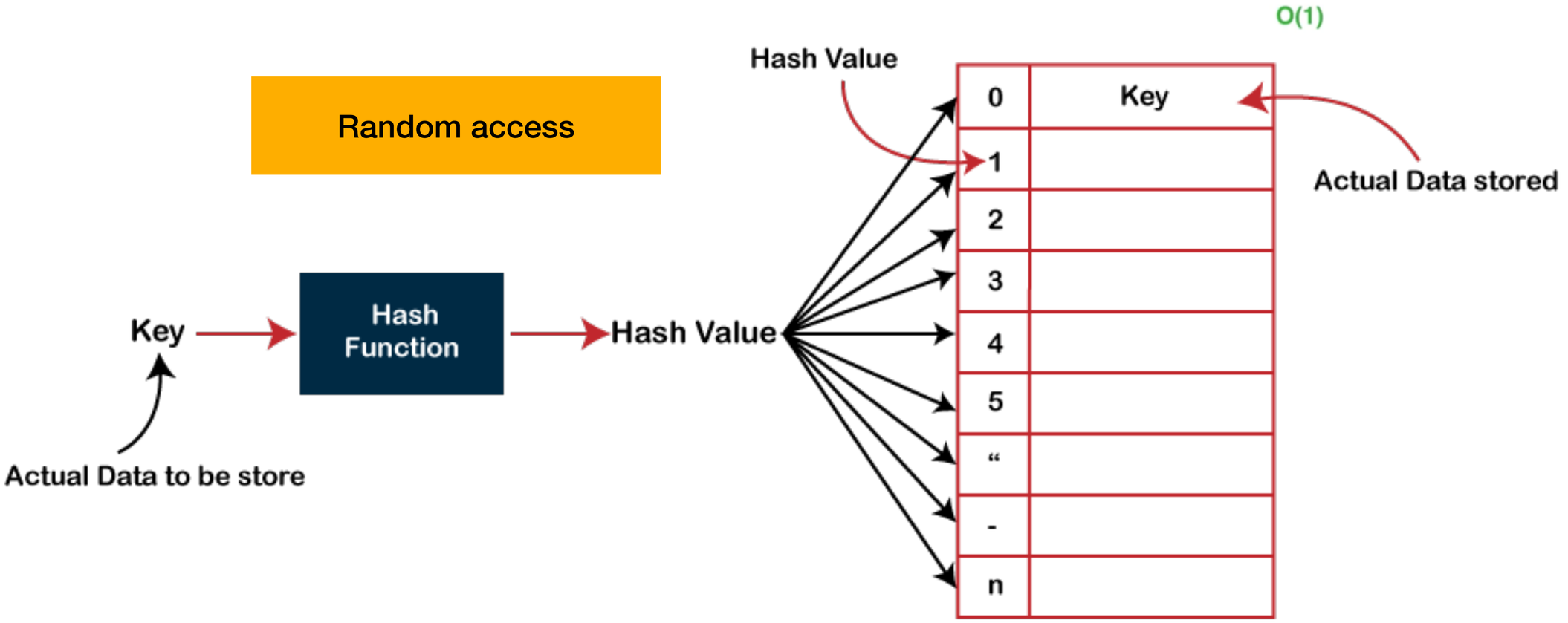
```
fun twoSum(nums: IntArray, target: Int): IntArray {
    for (i in nums.indices) {
        for (j in i + 1 until nums.size) {
            if (nums[i] + nums[j] == target) {
                return intArrayOf(i, j)
            }
        }
    }
    throw IllegalArgumentException("No two sum solution")
}
```

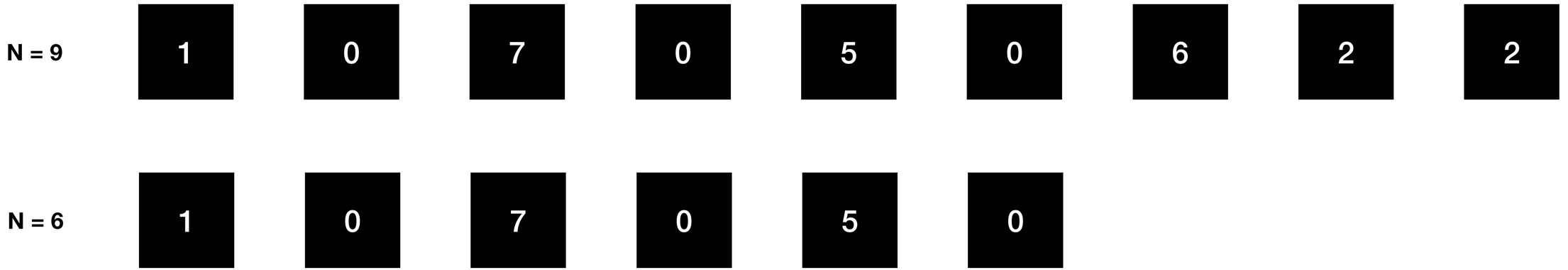
Algorithm solution:
Solving twosum in a single iteration

Sequential access



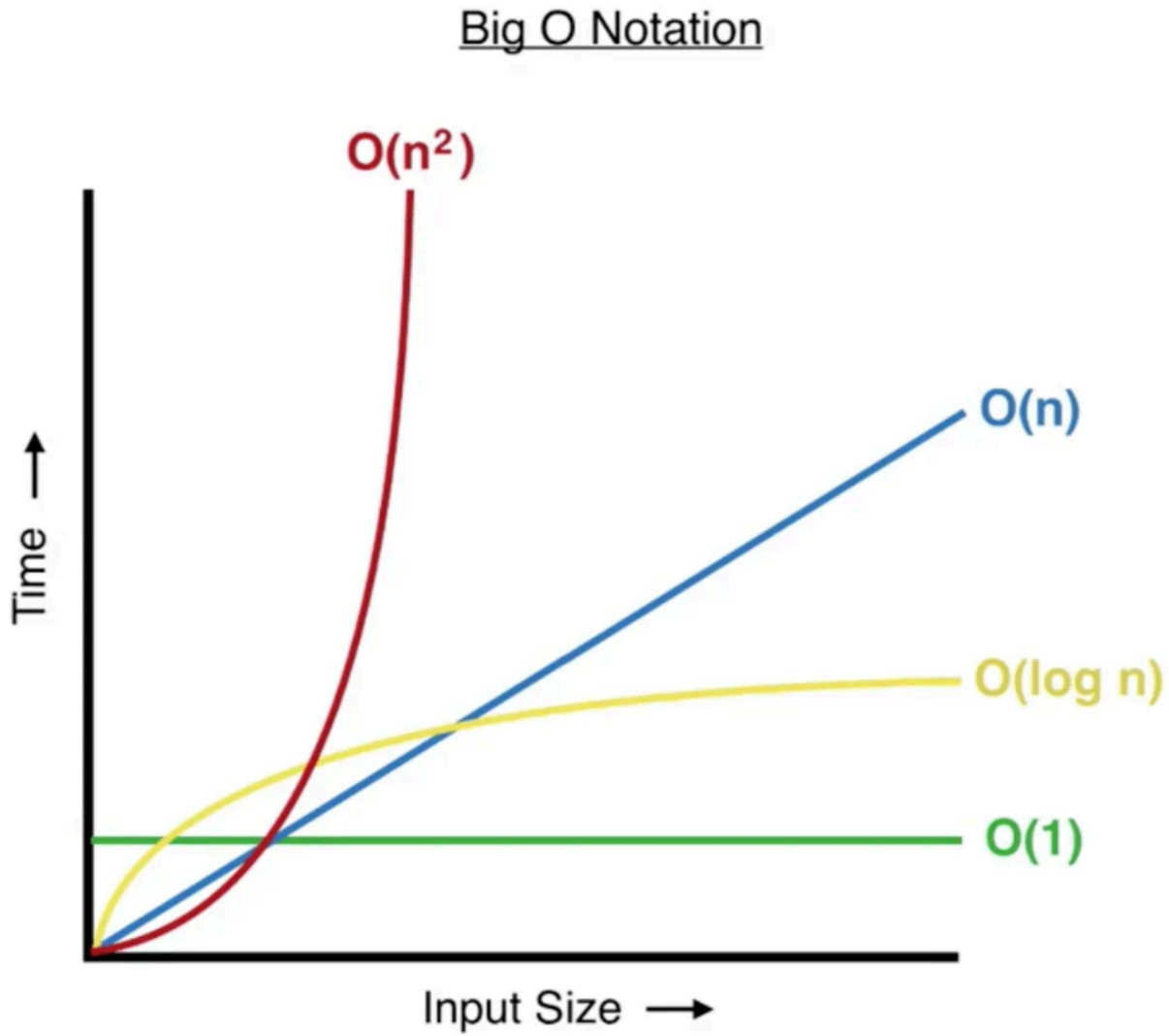
Random access





Runtime = O(6*9)

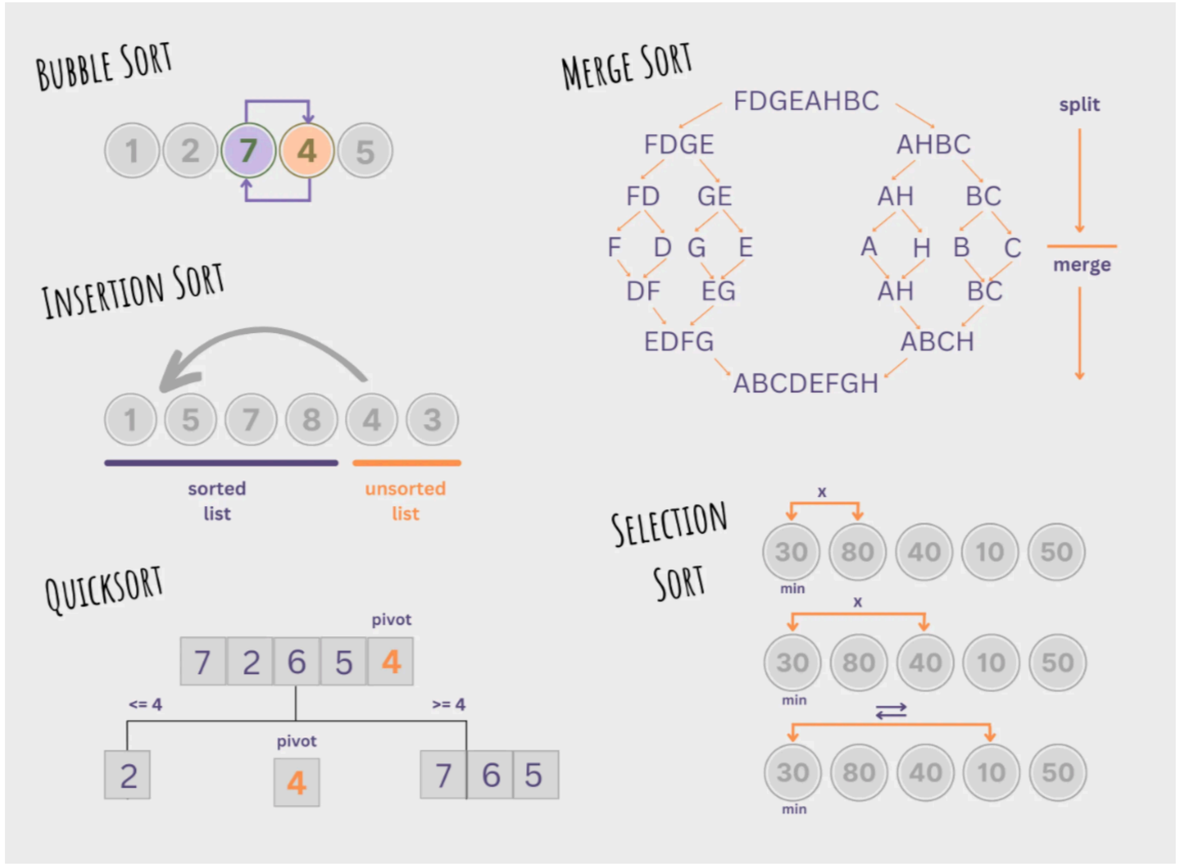
Big-O exercises



Sorting algorithms

CS101

- We will be working with 3 simple sorting algorithms
  - Bubble sort
  - Selection sort
  - Insertion sort
- You will be implementing selection sort & insertion sort in a pair programming fashion



Bubble-sort example