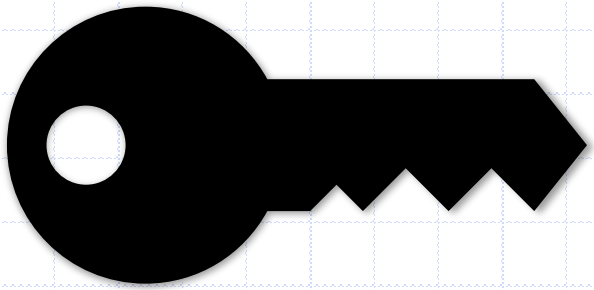
IT1312

Data Structures & Algorithms

Topic 03a: Search – Sequential Search

Introduction to Search Algorithms

- What is search?
- Experience losing a Key and finding it
- How to did you find it? How long did it take?



Learning Outcomes

- Describe Search Algorithms
- Identify types of Search algorithms
- Know the differences between unsorted and sorted list search
- Code Sequential Search
- Define the Complexity (Big-O) of Sequential Search

Search Algorithms

- Searching is the process of finding an item within a sequence using a **search key** to identify the specific item.

- **Sequential Search** (or Linear Search)
  - Unsorted List
  - Sorted List
- **Binary Search (*Next Topic*)**

Sequential Search

- Also known as Linear Search
  - Compares each item in a list of value against a target value (search key)
  - The search terminates when match is found or list is exhausted (not found)
  - For a successful search of n records
    - the best time is 1 comparison
    - the worst time is n comparison

Sequential Search

- **Algorithm:**
  - Start with the first item in the list
  - Compare it against the target value
  - If a match is found, return position in the list containing the target value
  - Else move to next item in the list and repeat the comparison

Sequential Search

10	7	1	3	-4	2	20

The array we are searching. Lets search for the number 3.

3?

10	7	1	3	-4	2	20

Start at the beginning and check the first element.

Is it 3?

http://www.sparknotes.com

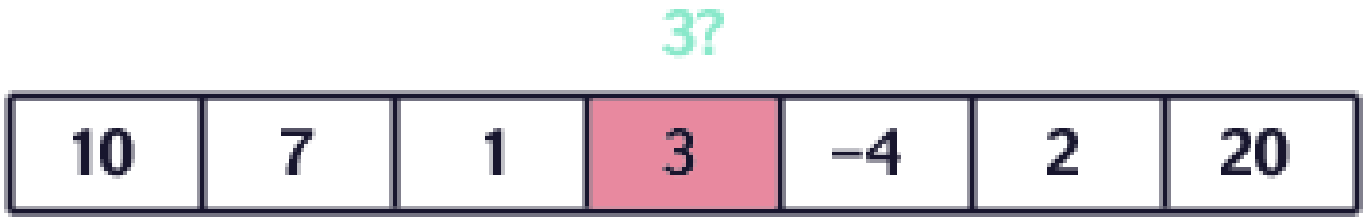
Sequential Search

No, not it. Is it the next element?

Not there either. The next element?

http://www.sparknotes.com

Sequential Search



We found it !!!

http://www.sparknotes.com

Sequential Search

```
def sequentialSearch( theValues, target ):
    n = len( theValues )

    for i in range(n):
        # If the target is in the ith element, return True
        if theValues[i] == target:
            return True

    return False     # If not found, return False
```

What if the list is already sorted?

Can we improve on the implementation with fewer comparisons?

(Sorted) Sequential Search

```
def sortedSequentialSearch( theValues, target ):
    n = len( theValues )

    for i in range(n):
        # If the target is in the ith element, return True
        if theValues[i] == target:
            return True
        elif _____________________:
            _________________

    return False     # If not found, return False
```

Complete the codes.

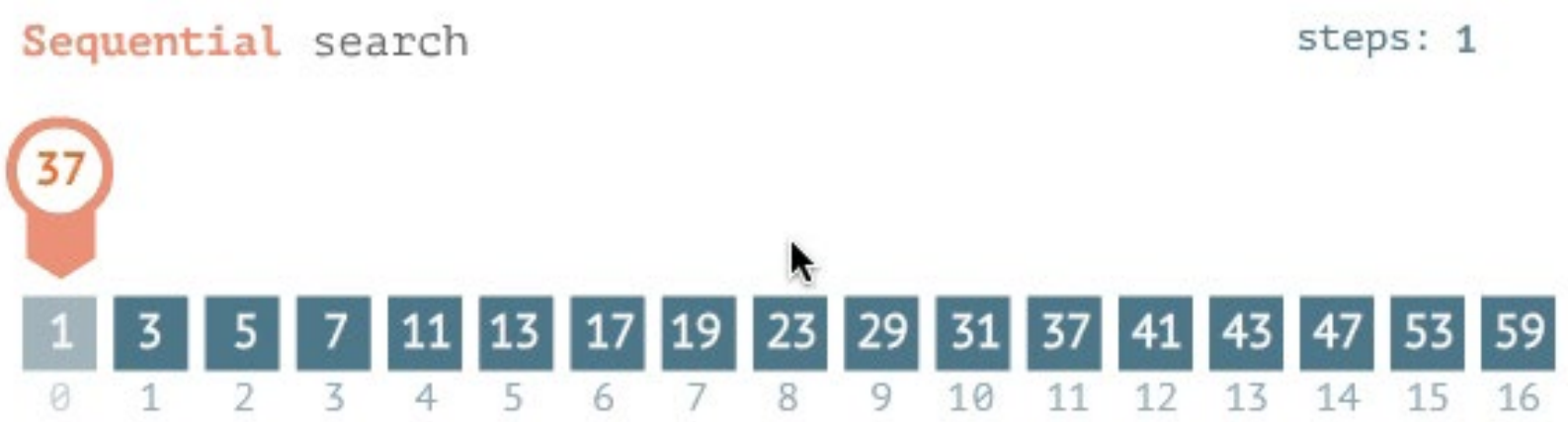
(Sorted) Sequential Search

```
def sortedSequentialSearch( theValues, target ):
    n = len( theValues )

    for i in range(n):
        # If the target is in the ith element, return True
        if theValues[i] == target:
            return True
        elif theValues [i] > target: #STOP searching
            return False


    return False     # If not found, return False
```

Assumption : Sorted in Ascending order

More examples

Here is a list of sorted numbers. Find key 37.
It takes 11 steps.
If the key is NOT there, it will stop when the number in the list is greater than the key



Reference:
https://www.freecodecamp.org/news/the-complexity-of-simple-algorithms-and-data-structures-in-javascript-11e25b29de1e/

Complexity

- Complexity (O) is a factor involved in a complex process.
- In algorithms & data structures, Time Complexity is the time required to perform a specific task (search, sort or access data) on a given data structure.
- Efficiency of performing a task is dependent on the number of operations required to complete a task.

Sequential Search

- Complexity (O)
  - Best case for searching an item in a sorted list, one after another, is a constant O(1) where **O** is **Operation Count (Big-O)**
  - The complexity of your search is constant with the list size. The average to the worst case of this kind of search is a linear complexity or O(n).
  - In other words, for n items, you have to look n times, before you find your item

Summary

- Sequential Search (Linear Search)
- Simplest and easy to implement
- But it is not very efficient
- Next, we will explore another search algorithm – Binary Search