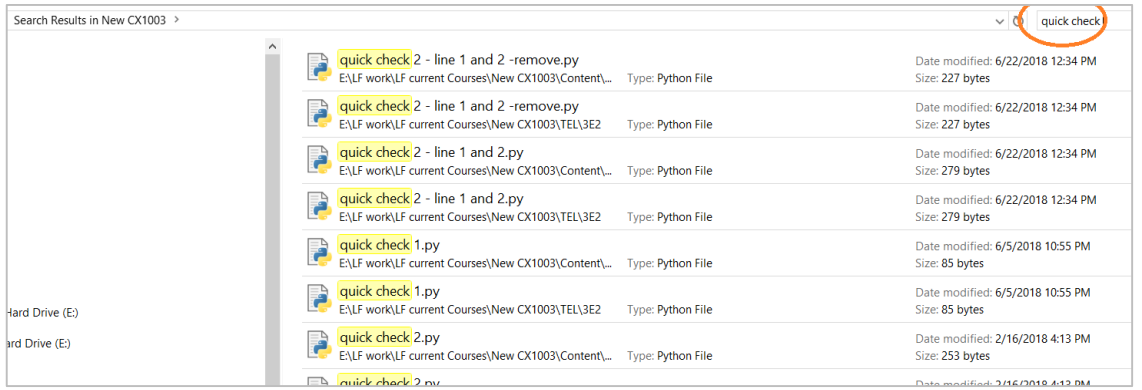
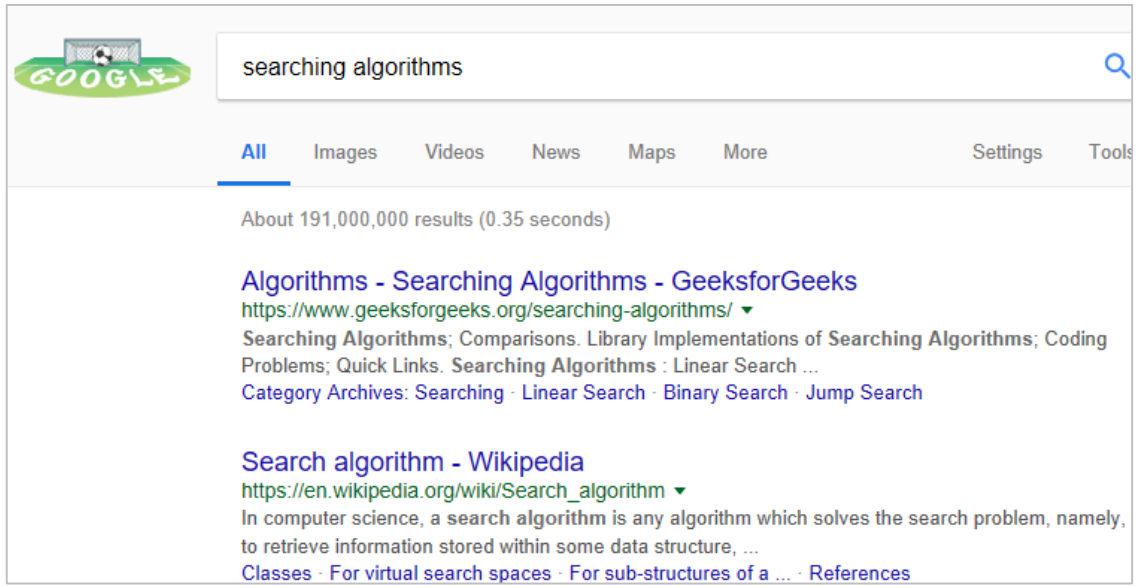
Algorithm Design: Searching

Searching


SEARCHING


Search Results in New CX1003
quick check
quick check 2 - line 1 and 2 -remove.py
E:\LF work\LF current Courses\New CX1003\Content\... Type: Python File
Date modified: 6/22/2018 12:34 PM
Size: 227 bytes
quick check 2 - line 1 and 2 -remove.py
E:\LF work\LF current Courses\New CX1003\TEL\3E2 Type: Python File
Date modified: 6/22/2018 12:34 PM
Size: 227 bytes
quick check 2 - line 1 and 2.py
E:\LF work\LF current Courses\New CX1003\Content\... Type: Python File
Date modified: 6/22/2018 12:34 PM
Size: 279 bytes
quick check 2 - line 1 and 2.py
E:\LF work\LF current Courses\New CX1003\TEL\3E2 Type: Python File
Date modified: 6/22/2018 12:34 PM
Size: 279 bytes
quick check 1.py
E:\LF work\LF current Courses\New CX1003\Content\... Type: Python File
Date modified: 6/5/2018 10:55 PM
Size: 85 bytes
quick check 1.py
E:\LF work\LF current Courses\New CX1003\TEL\3E2 Type: Python File
Date modified: 6/5/2018 10:55 PM
Size: 85 bytes
quick check 2.py
E:\LF work\LF current Courses\New CX1003\Content\... Type: Python File
Date modified: 2/16/2018 4:13 PM
Size: 253 bytes
Hard Drive (E:)
ard Drive (E:)


GOOGLE
searching algorithms
All
Images
Videos
News
Maps
More
Settings
Tools
About 191,000,000 results (0.35 seconds)
Algorithms - Searching Algorithms - GeeksforGeeks
https://www.geeksforgeeks.org/searching-algorithms/
Searching Algorithms; Comparisons. Library Implementations of Searching Algorithms; Coding Problems; Quick Links. Searching Algorithms : Linear Search ...
Category Archives: Searching · Linear Search · Binary Search · Jump Search
Search algorithm - Wikipedia
https://en.wikipedia.org/wiki/Search_algorithm
In computer science, a search algorithm is any algorithm which solves the search problem, namely, to retrieve information stored within some data structure, ...
Classes · For virtual search spaces · For sub-structures of a ... · References


196 B/s
6:43 PM
53%
Lee
Cancel
Lee Hao Dong
+65 9385 0229
Valeria Lee
+65 9324 9231
Search

Lesson Objectives

At the end of this lesson, you should be able to:

- Describe the process of searching
- Explain the importance of different types of searching algorithms
- Search for a given value in an array using linear search and binary search
- Apply search algorithms in problem solving
- Recognize that “no single” best search algorithm applies to all scenarios

Topic Outline

- Importance of Searching
- Linear Search
- Binary Search
- Comparison of Searching Algorithms

Searching Pizza Hut in North Spine Plaza

- Starts at the first item
- Is it the one I am looking for?
- If not, go to next item
- Repeats until found or all the items are checked

Linear Search

- Iterates over the sequence, one item at a time, until the specific item is found or all items have been examined
  - The element that needs to be found is called a search key
- Linear search/ sequential search
  - Intuitive approach
  - Starts at the first item
  - Is it the one I am looking for?
  - If not, go to next item
  - Repeats until found or all the items are checked
- This approach is necessary if items are not sorted

Searching in a Sorted List

List of Food & Beverage in **North Spine**

Bakery Cuisine

Mr Bean

North Spine Food Court

Peach Garden Chinese Restaurant

Pizza Hut

Subway

The Soup Spoon Union

If items are sorted

NOT HERE

Searching in a Sorted List (Cont'd)

If items are sorted

List of Food & Beverage in **North Spine**

- Bakery Cuisine
- Mr Bean
- North Spine Food Court
- Peach Garden Chinese Restaurant
- Pizza Hut
- Subway
- The Soup Spoon Union

NOT HERE

Searching

- Given a list of data, **searching** is finding the location of a particular value or reporting that the value is not present.
- It is one of the fundamental problems in computer science and programming.
- Sorting is done to make searching easier.
- There are multiple searching algorithms to solve problems.
  - How do we know which algorithm is better?

Searching in a Dictionary

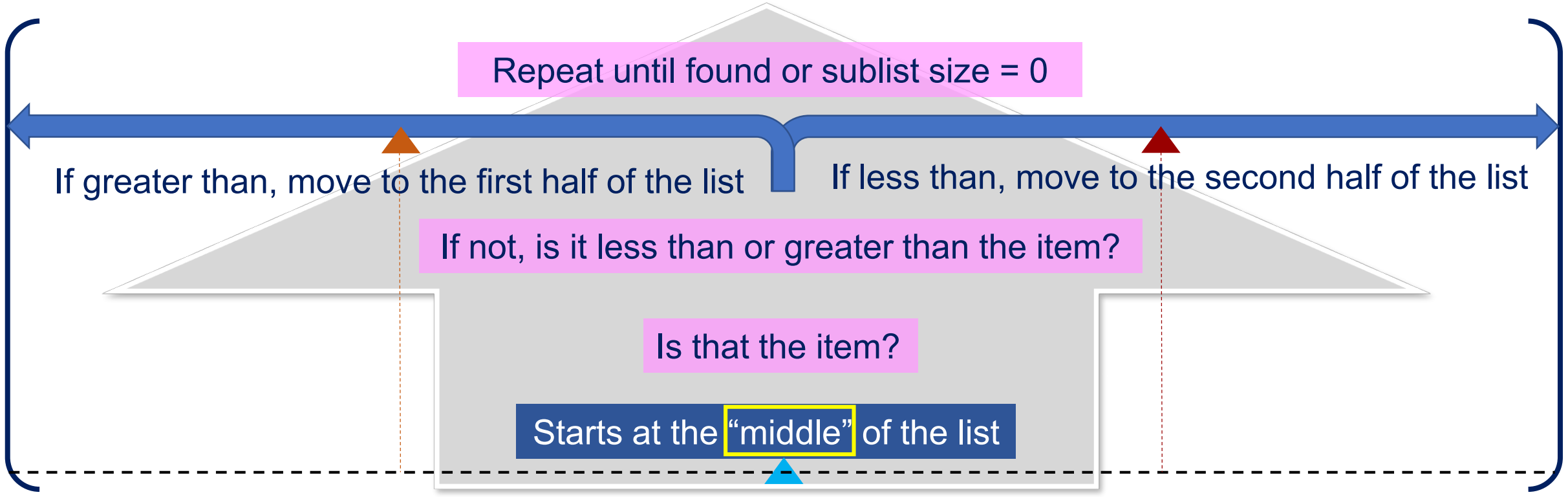
Binary Search

Binary Search on List of Ascending Order

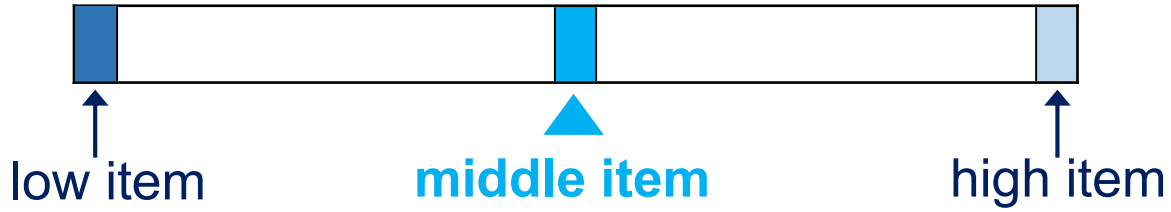
If items are sorted then you can divide-and-conquer

- *dividing your work in half with each step*

Generally a good thing!



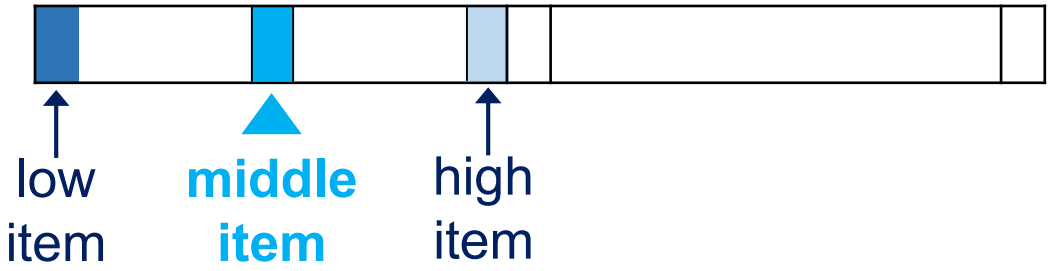
Binary Search Illustration

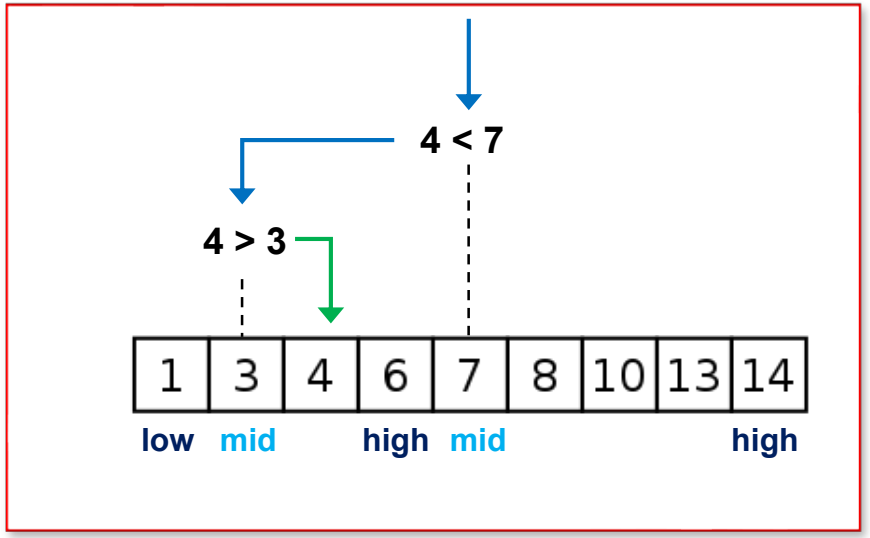


Is the middle item what we are looking for?

If not, is it lower or greater than the target item?

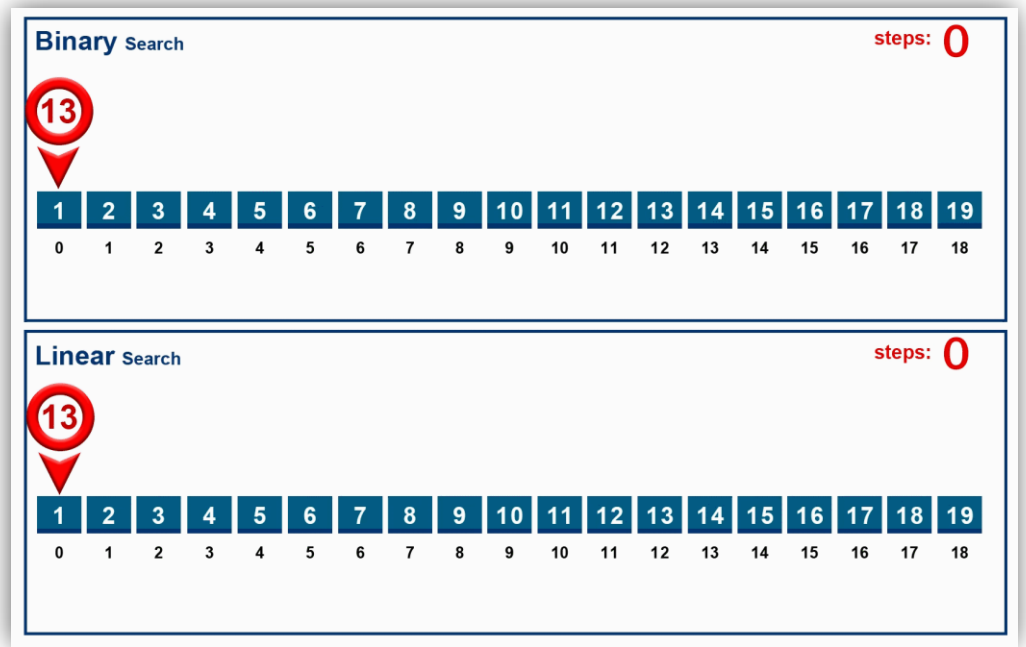
(**Assume lower**)





Linear Search vs. Binary Search

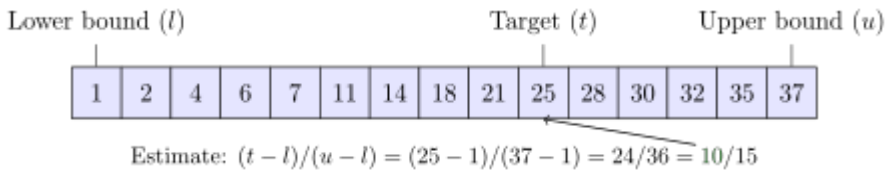
Variance of Hi-Low Number Guessing Game



Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)

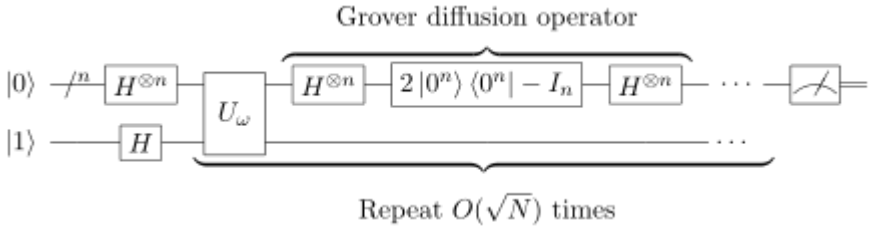
Other Searching Algorithm

Interpolation Search



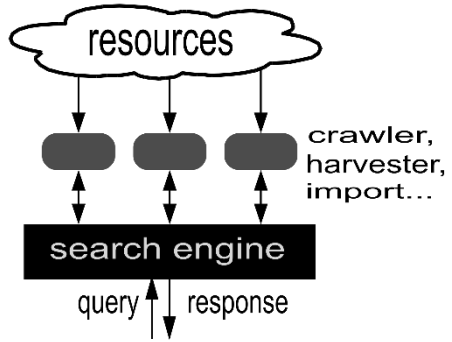
more like what people really do

Grover's Algorithm

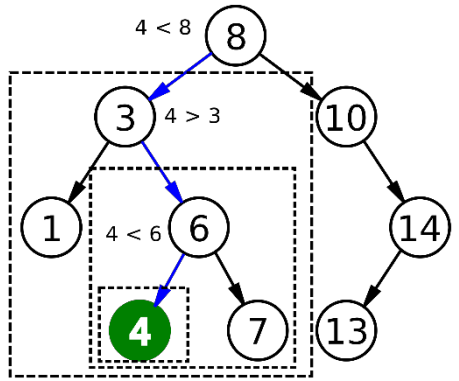


waiting for quantum computers to be built

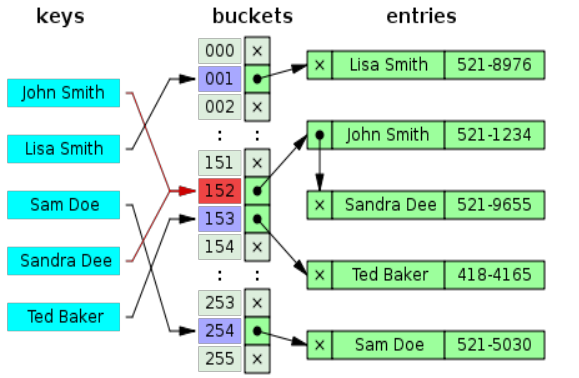
Indexed Searching



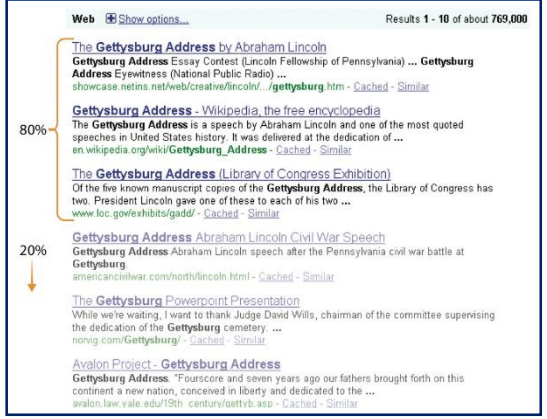
Binary Search Trees



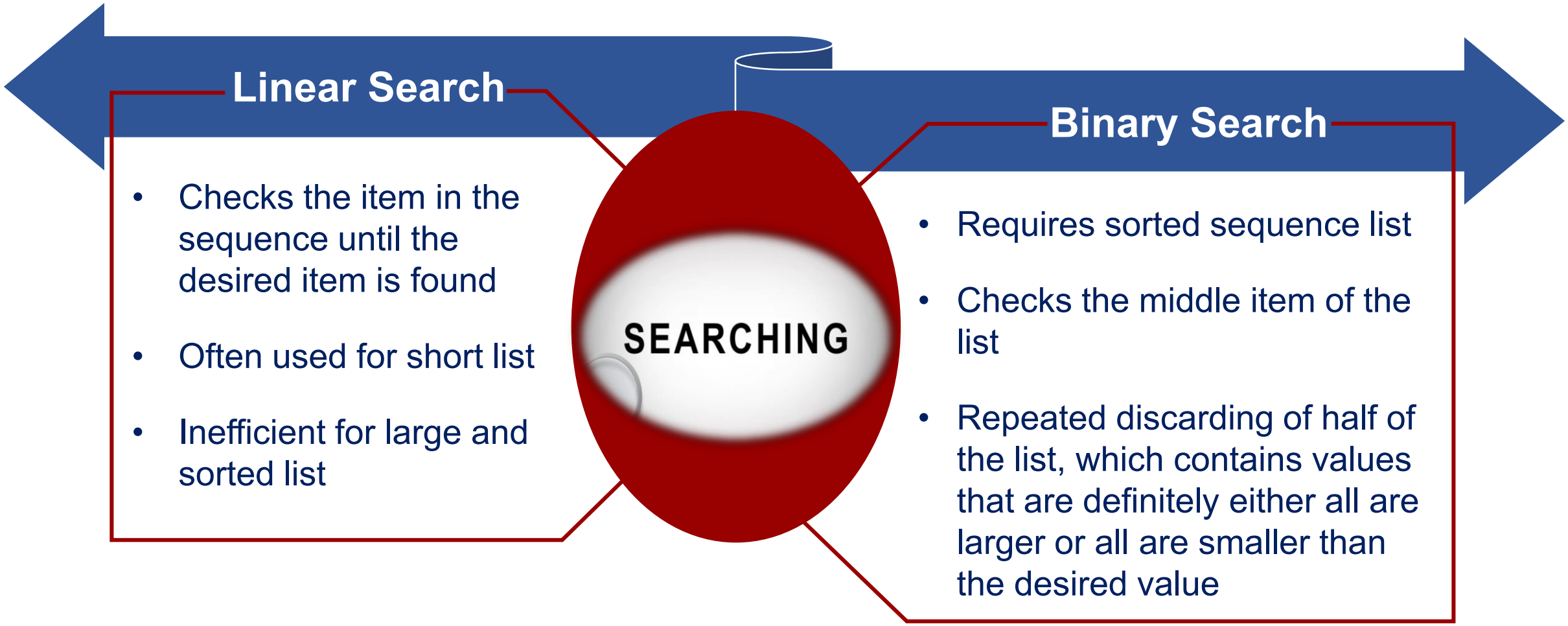
Hash Table Searching



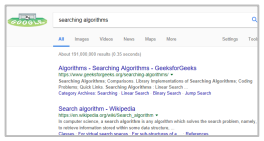
Best-first



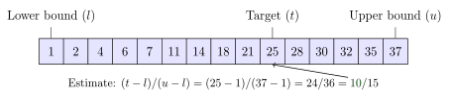
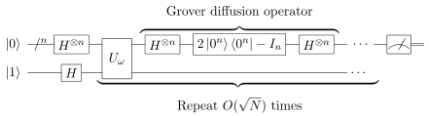
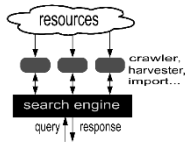
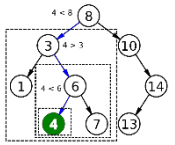
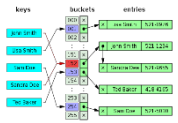
Summary


Linear Search
Checks the item in the sequence until the desired item is found
Often used for short list
Inefficient for large and sorted list
SEARCHING
Binary Search
Requires sorted sequence list
Checks the middle item of the list
Repeated discarding of half of the list, which contains values that are definitely either all are larger or all are smaller than the desired value

References for Images

No.	Slide No.	Image	Reference
1	3		Retrieved July 16, 2018 from from https://www.google.com/.
2	7		By Source, Fair use, retrieved July 16, 2018 from https://en.wikipedia.org/w/index.php?curid=22312809.
3	7, 8, 9		Magnifying Glass [Online Image]. Retrieved July 16, 2018 from http://www.publicdomainfiles.com/show_file.php?id=13534684215801.
4	8, 9		By Burger King - https://bk.com, Public Domain, retrieved July 16, 2018 from https://commons.wikimedia.org/w/index.php?curid=53835318.
5	10		Magnifier [Online Image]. Retrieved July 16, 2018 from https://pixabay.com/en/copyright-magnifier-magnifying-glass-389901/.

References for Images

No.	Slide No.	Image	Reference
6	14		By Esquivalience - Own work, CC0, retrieved July 16, 2018 from https://commons.wikimedia.org/w/index.php?curid=63614695.
7	14		By Bender2k14 - Created in LaTeX code using Q-circuit. Source code follows this template., CC BY-SA 3.0, retrieved July 16, 2018 from https://commons.wikimedia.org/w/index.php?curid=13524800.
8	14		By Jakob Voss - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=3112366.
9	14		By Binary_search_tree.svg: Booyabazookaderivative work: movax - Binary_search_tree.svg, Public Domain, retrieved July 16, 2018 from https://commons.wikimedia.org/w/index.php?curid=11079967.
10	14		By Jorge Stolfi - Own work, CC BY-SA 3.0, retrieved July 16, 2018 from https://commons.wikimedia.org/w/index.php?curid=6471915.

References for Images

No.	Slide No.	Image	Reference
11	14		Morville, P. (2010). 4-10 Best First with Google [Online Image]. Retrieved July 16, 2018 from https://www.flickr.com/photos/morville/4274337684/.

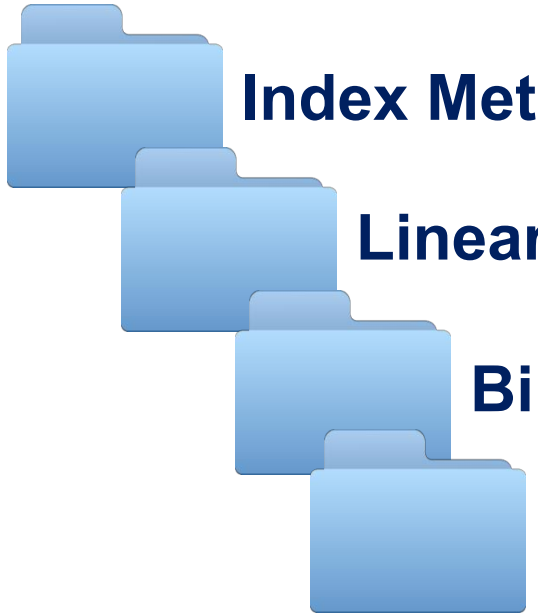
Algorithm Implementation: Searching in Python

Lesson Objectives

At the end of this lesson, you should be able to:

- Use index method in Python
- Explain the importance of coding searching algorithms in Python
- Code linear search in Python
- Code binary search in Python
- Identify other search algorithms written in Python
- Apply your knowledge and understanding of searching algorithms to your problem solving in Python

Topic Outline



- Index Method in Python
- Linear Search
- Binary Search
- Comparison of Searching Algorithms

Linear Search: Recall

- Iterates over the sequence, one item at a time, until the specific item is found or all items have been examined
  - The element that needs to be found is called a search key
- Linear search/ sequential search
  - Intuitive approach
  - Starts at the first item
  - Is it the one I am looking for?
  - If not, goes to next item
  - Repeats until found or all the items are checked
- This approach is necessary if items are not sorted

Linear Search in Python

```
foodList = ['Bakery Cuisine', 'Subway', 'Peach Garden Chinese Restaurant',
            'Mr Bean', 'Pizza Hut', 'The Soup Spoon Union',
            'North Spine Food Court']

for item in foodList:
    if item == 'Pizza Hut':
        print('Pizza Hut is in list')
        break
```

Python List `index()` Method

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Description

The method `index()` returns the lowest index in list that ***obj*** appears.

The **index** method does a linear search and stops at the first matching item.

If no matching item is found, it raises a **ValueError** exception.

Python List `index()` Method: Example

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

```python
foodList = ['Bakery Cuisine', 'Subway', 'Peach Garden Chinese Restaurant',
            'Mr Bean', 'Pizza Hut', 'The Soup Spoon Union',
            'North Spine Food Court']

print ("Index for Pizza Hut : ", foodList.index( 'Pizza Hut' ))
print ("Index for Pizza Hut : ", foodList.index( 'pizza hut' ))
```

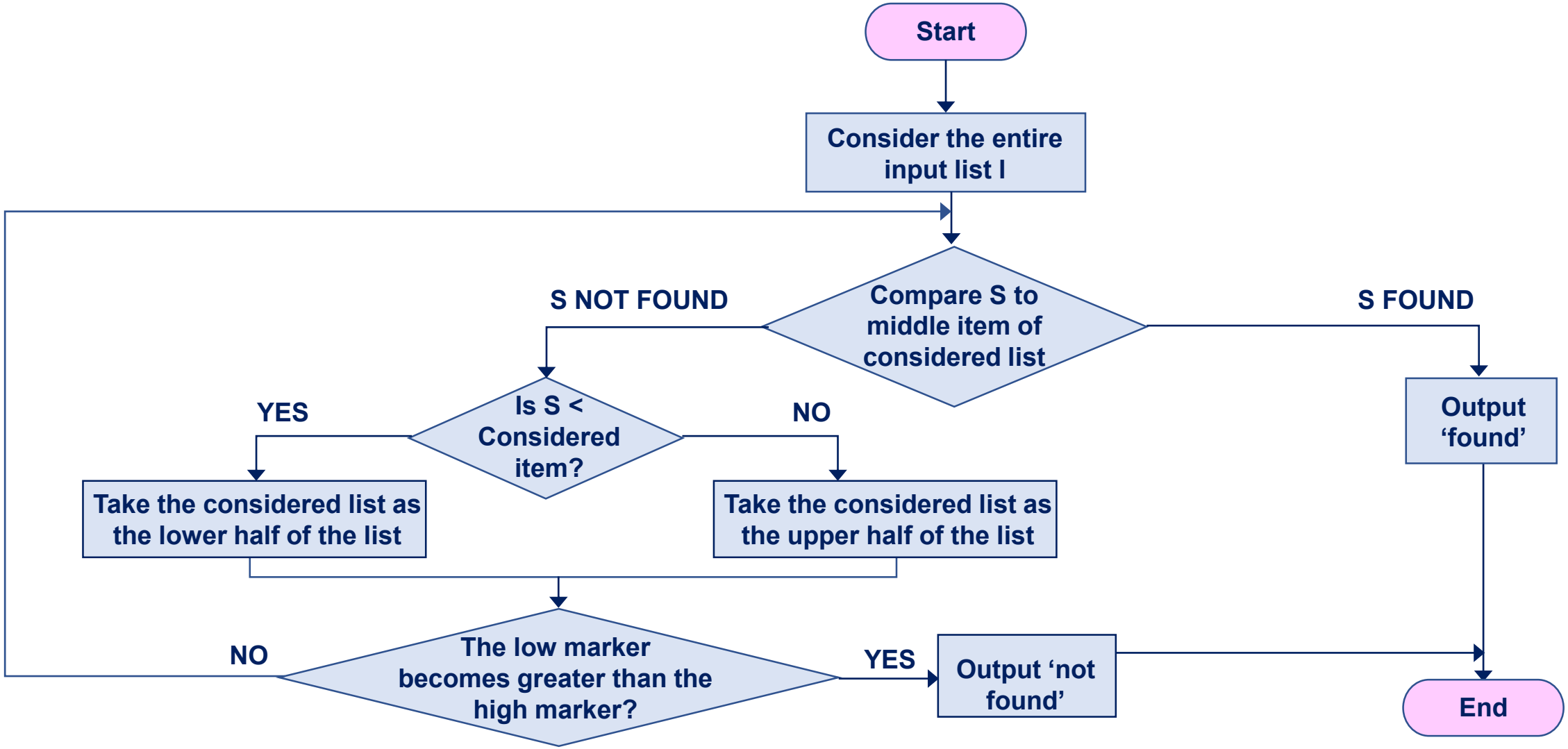
(Index labels: 'Bakery Cuisine' 0, 'Subway' 1, 'Peach Garden Chinese Restaurant' 2, 'Mr Bean' 3, 'Pizza Hut' 4)

```
Index for Pizza Hut :  4
Traceback (most recent call last):
  File "C:/New CX1003/7B2/indexPizzahut.py", line 6, in <module>
    print ("Index for Pizza Hut : ", foodList.index( 'pizza hut' ))
ValueError: 'pizza hut' is not in list
```

This method returns index of the found object.

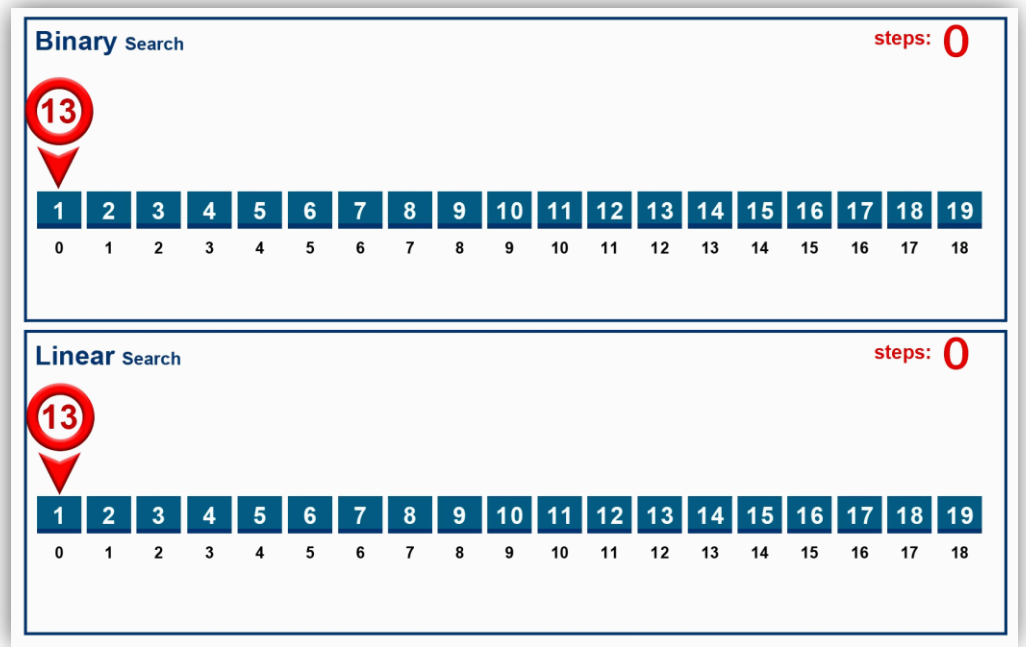
Otherwise, it raises an exception indicating that the value is not found.

Binary Search Flowchart



Linear Search vs. Binary Search

Variance of Hi-Low Number Guessing Game



Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)

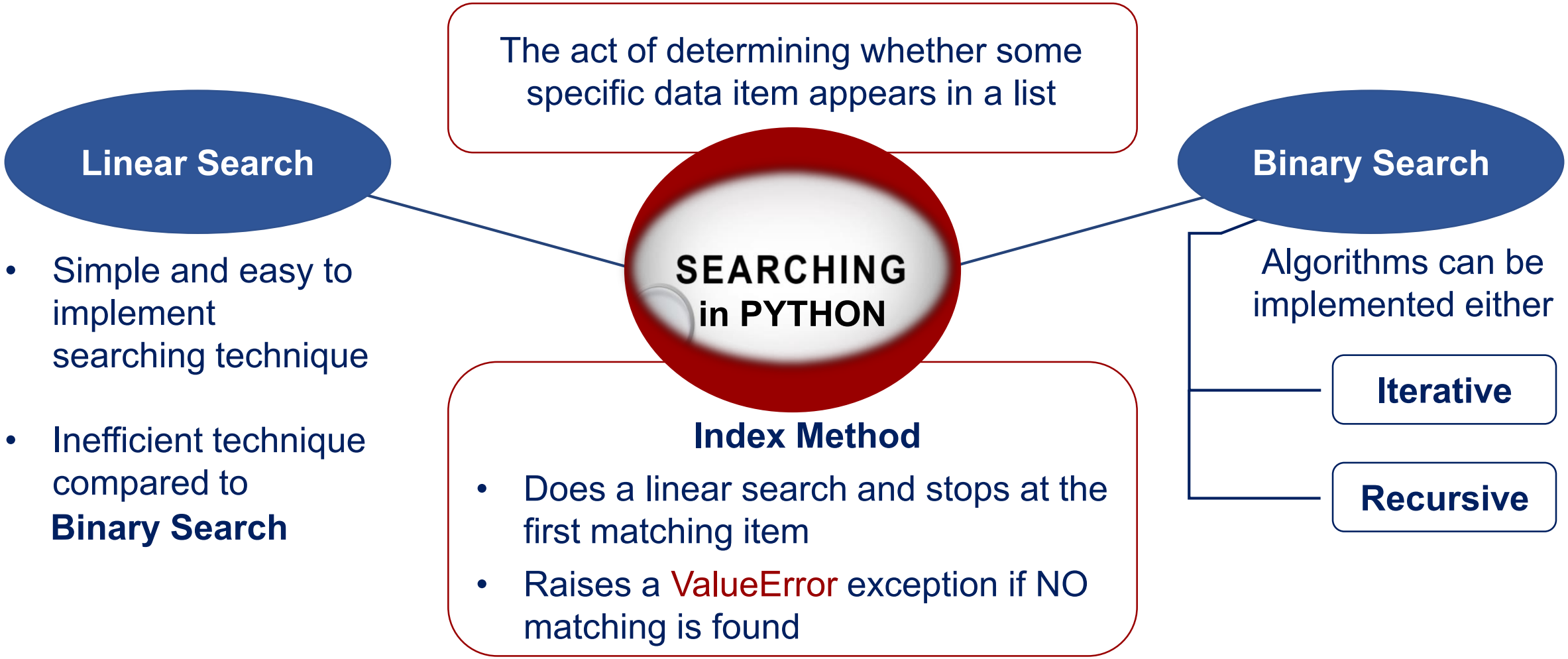
Iterative Binary Search

```
def binarySearch( items, target ) :
      # Start with the entire list
    low = 0
    high = len(items) - 1
      # Repeatedly subdivide the list in half until the target is found
    while low <= high :
        # locates the middle item of the list
        mid = (low + high) // 2
        # compares middle item with the search key
        if items[mid] == target:
            return True
        # target is less than middle item?
        elif target < items[mid] :
            high = mid - 1
        # target is greater than middle item?
        else :
            low = mid + 1
    return False

numbers = range(1, 20, 1)
search_key = 7
if (binarySearch( numbers, search_key )):
    print (search_key, "is in the list")
else:
    print (search_key, "is not in the list")
```

Recursive Binary Search

```
def binary_search(items, target, low = 0, high = None):
    if high == None:
        high = len(items) - 1

    if low > high:
        return False

    mid = (low + high) // 2

    if target == items[mid]:
        return True
    elif target > items[mid]:
        return binary_search(items, target, low = (mid + 1), high = high)
    else:
        return binary_search(items, target, low = low, high = (mid-1))


numbers = range(1, 50, 1)
search_key = 34
if (binary_search( numbers, search_key, 0 )):
    print (search_key, "is in the list")
else:
    print (search_key, "is not in the list")
```

Summary


The act of determining whether some specific data item appears in a list
Linear Search
Binary Search
SEARCHING in PYTHON
• Simple and easy to implement searching technique
• Inefficient technique compared to **Binary Search**
Algorithms can be implemented either
Iterative
Recursive
Index Method
• Does a linear search and stops at the first matching item
• Raises a ValueError exception if NO matching is found

References for Images

No.	Slide No.	Image	Reference
1	5		By Source, Fair use, retrieved July 16, 2018 from https://en.wikipedia.org/w/index.php?curid=22312809.
2	5		Magnifying Glass [Online Image]. Retrieved July 16, 2018 from http://www.publicdomainfiles.com/show_file.php?id=13534684215801.
3	All pages with Python codes		Python Logo [Online Image]. Retrieved April 18, 2018 from https://pixabay.com/en/language-logo-python-2024210/.