

## Chapter 5.4: Linear Search

1 of 2

### Objectives

- Define a linear search
- Justify an advantage & disadvantage of a linear search
- Explain a suitable use (application) for a linear search

### INTRODUCTION

A linear search is a **sequential** search which simply means, starts at the beginning and works through each item in turn.

Linear searches are easy to code and the list you are searching does not need to be in order.

The algorithm (steps) of a linear search:

1. START at the first item
2. COMPARE the list item with the search criterion
3. IF they are the same then STOP
4. ELSE if they are not the same move on to the next item
5. REPEAT steps 2 - 4 until the end of the list OR the search criterion is found
6. OUTPUT result

### TASK 1

Type up the advantages and disadvantages of a linear search.

Advantages	Disadvantages
Easy to code	It can take a very long time to find a number as it checks each number at a time
The data doesn't need to be sorted before searching	If the number you're looking for doesn't exist, it will need to search through the full list

### TASK 2

[This video from Python School.Net shows](#) you how to write a linear search in Python. Build on this example and code the following.

- A program that stores 10 surnames in an array.
- The program should ask the user what name they want to try and find.
- The program should tell the user if the name exists and what location of the array the name is positioned.
- The program should also feedback if the name does not exist in the list.

Supply a link to your code in Repl.it below so I can check that it works or fully annotated code from IDLE.

## Chapter 5.4: Linear Search

2 of 2

```
def linearSearch(myItem, myList):
    found = False
    position = 0
    while position < len(myList) and not found:
        if myList[position] == myItem:
            found = True
            position = position + 1
    return found

if 1 == 1:
    surnames = ["Johnson", "Stevenson", "Smith", "Jones", "Davies", "Wilson", "Thompson",
               "Robinson", "Wood", "Jackson"]
    item = input("What surname do you want to find?")
    isItFound = linearSearch(item, surnames)
    if isItFound:
        print("Your surname you searched for was found.")
    else:
        print("The surname you searched for was not found.")
```

```
What surname do you want to find? Jones
Your surname you searched for was found.
```

<https://repl.it/@ManrajLally/Linear-Search> You can see I that at the top I define "linearSearch" which is basically making it check everything in a list as the position of the element it checks moves up by one every time. I couldn't use if \_\_name\_\_ == "\_\_main\_\_" because I'm using repl.it so instead I just used if 1 == 1. Then you see my list of surnames and an input is needed to search. If it's found, the output is "it's found", if it's not found, "it wasn't found".

### TASK 3:

**What is the worst case scenario in a list of 100 items if the search criterion does not exist?**

Firstly, your item won't be found and secondly it will probably take a while to search a 100 items. Basically it's a large waste of time.

### Extension

**Extend the program to allow the user to enter new names into the array.**

Supply a link to your code in Repl.it below so I can check that it works or fully annotated code from

```
def linearSearch(myItem, myList):
    found = False
    position = 0
    while position < len(myList) and not found:
        if myList[position] == myItem:
            found = True
            position = position + 1
    return found

if 1 == 1:
    surnames = ["Johnson", "Stevenson", "Smith", "Jones", "Davies", "Wilson", "Thompson",
               "Robinson", "Wood", "Jackson"]
    moreSurnames = input("Type the surname you wish to add.")
    surnames.append(moreSurnames)
    item = input("What surname do you want to find?")
    isItFound = linearSearch(item, surnames)
    if isItFound:
        print("Your surname you searched for was found.")
    else:
        print("The surname you searched for was not found.")
```

IDLE

```
Type the surname you wish to add. bob
What surname do you want to find? bob
Your surname you searched for was found.
```