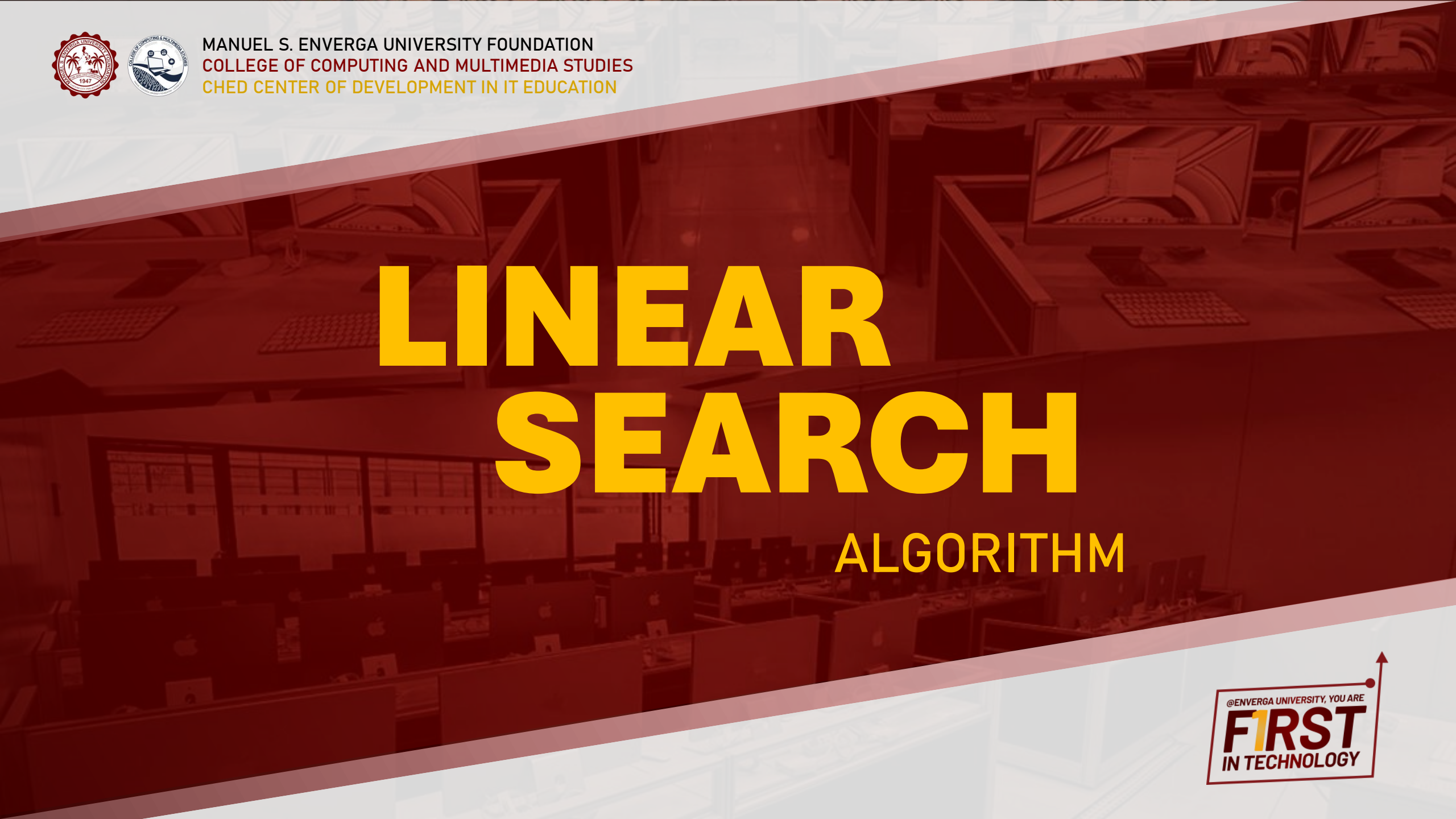
MANUEL S. ENVERGA UNIVERSITY FOUNDATION
COLLEGE OF COMPUTING AND MULTIMEDIA STUDIES
CHED CENTER OF DEVELOPMENT IN IT EDUCATION



LINEAR SEARCH

ALGORITHM

@ENVERGA UNIVERSITY, YOU ARE FIRST IN TECHNOLOGY

GROUP 1

MEMBERS

AMREI LOUIS A. ESPARES

JHORAM EZEKIEL C. BALA

DANIEL RICHARD S. GUINTO

PETER BOB R. DOMOGMA

@ENVERGA UNIVERSITY, YOU ARE FIRST IN TECHNOLOGY

Learning Outcomes

- Define linear search algorithm
- Recognize the limitations of linear search and compare it to more efficient algorithms like binary search, especially in sorted datasets.
- Recognize Limitations: Understand why linear search may not be suitable for large datasets or frequently repeated searches
- Write a function to search for a specific value in an array
- Implement search functionality in a real-world project
- Learn how the linear search algorithm sequentially checks each element to find a match.

What is **Linear Search?**

Linear search algorithm is the simplest searching algorithm that is used to find an element in the given collection.

It simply compares the element to find with each element in the collection one by one till the matching element is found or there are no elements left to compare.

Applications of Linear Search

- **Unsorted Lists:** When we have an unsorted array or list, linear search is most commonly used to find any element in the collection.
- Linear search is commonly used in datasets where the data is unsorted or unordered.

Example: Searching for a specific book in a library database where the records are not sorted alphabetically.

Applications of Linear Search

- **Small Data Sets:** Linear Search is preferred over binary search when we have small data sets.
- Ideal for searching in small datasets, where its simplicity outweighs the inefficiency of O(n)O(n)O(n) complexity.

Example: Finding an item in a short list of customer preferences or recently accessed files.

Applications of Linear Search

- **Searching Linked Lists:** In linked list implementations, linear search is commonly used to find elements within the list. Each node is checked sequentially until the desired element is found.
- For linked lists, where random access is not possible, linear search is often used to traverse the list to find a target element.

Example: A list of students, each containing details like ID, name, and grade, is stored in a linked list. Linear search is used to find a specific student by ID or name, and Checking if a student exists before adding new grades.

Applications of Linear Search

- **Simple Implementation:** Linear Search is much easier to understand and implement as compared to Binary Search or Ternary Search.
- Used in teaching basic search algorithms due to its simple and intuitive implementation.

Students can learn algorithm design and efficiency concepts through linear search.

Advantages of Linear Search

- Linear search can be used irrespective of whether the array is sorted or not. It can be used on arrays of any data type.
- Does not require any additional memory.
- It is a well-suited algorithm for small datasets.

Disadvantages of Linear Search

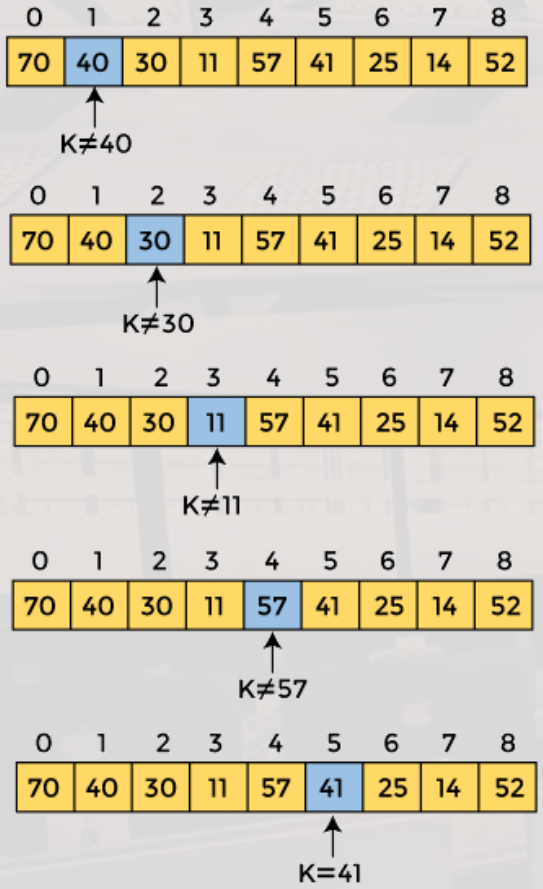
- Linear search has a time complexity of O(N), which in turn makes it slow for large datasets.
- Not suitable for large arrays.

Pseudocode Algorithm

```
//
//
// func doLinearSearch (container, target):
//     for index := 0 to container.end do:
//         if container[index] == targetIndex:
//             return index
//
//     else: error
//
//
//
```

```
// std::getline(std::cin, dummy);
// system("cls");
// displayAnimatedLinearSearch(numberLinkedList, 5);
// std::getline(std::cin, dummy);
// system("cls");
// displayAnimatedLinearSearch(numberLinkedList, 6);
// std::getline(std::cin, dummy);
// system("cls");
// displayAnimatedLinearSearch(numberLinkedList, 10);
// std::getline(std::cin, dummy);
```

Examples of Linear Search



list 65 20 10 55 32 12 50 99

search element 12

Step 1:

search element (12) is compared with first element (65)

list 65 20 10 55 32 12 50 99

12

Both are not matching. So move to next element

Step 2:

search element (12) is compared with next element (20)

list 65 20 10 55 32 12 50 99

12

Both are not matching. So move to next element

Step 3:

search element (12) is compared with next element (10)

list 65 20 10 55 32 12 50 99

12

Both are not matching. So move to next element

Step 4:

search element (12) is compared with next element (55)

list 65 20 10 55 32 12 50 99

12

Both are not matching. So move to next element

Step 5:

search element (12) is compared with next element (32)

list 65 20 10 55 32 12 50 99

12

Both are not matching. So move to next element

Step 6:

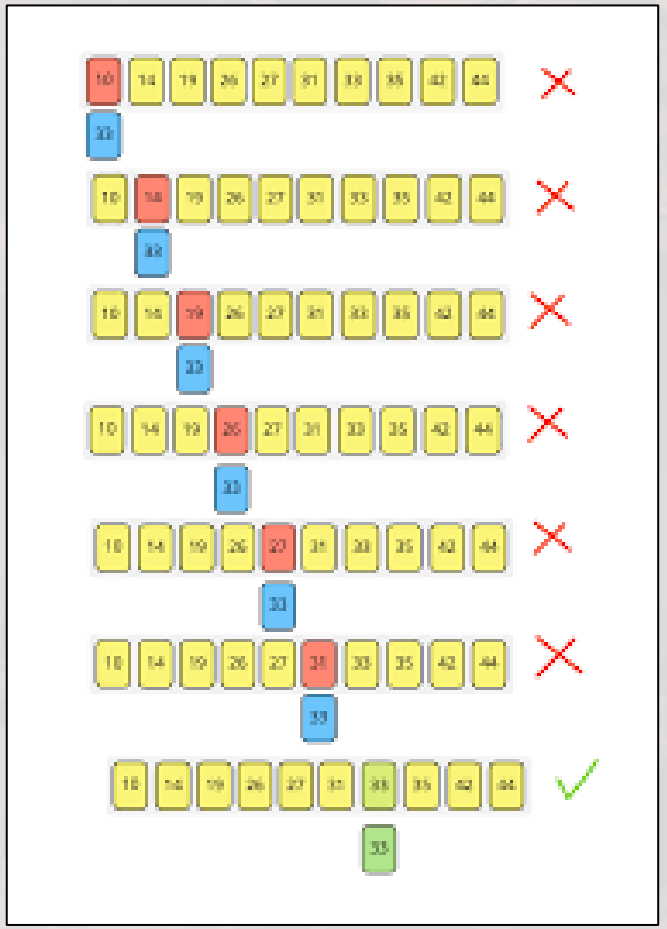
search element (12) is compared with next element (12)

list 65 20 10 55 32 12 50 99

12

Both are matching. So we stop comparing and display element found at index 5.

Examples of Linear Search



Source Code Implementation

https://github.com/Allie1080/linear-search-datastruct

MANUEL S. ENVERGA UNIVERSITY FOUNDATION
COLLEGE OF COMPUTING AND MULTIMEDIA STUDIES
CHED CENTER OF DEVELOPMENT IN IT EDUCATION

THANK YOU!

@ENVERGA UNIVERSITY, YOU ARE FIRST IN TECHNOLOGY