|  |  |
| --- | --- |
| **Data Structures & Algorithms**  Diploma in IT, CSF  Year 2 (2024/25) Semester 4 | **Week 8** |
| **1 Hour** |
| **Tutorial 8 – Searching Techniques** | |

1. Given an ***unsorted array*** of numbers below,

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 48 | 25 | 95 | 76 | 57 | 12 | 33 | 88 | 63 | 82 |

1. How many comparisons are required to search for 57?

5

1. How many comparisons are required to search for 35?

10

1. Given a ***sorted array*** of numbers below,

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 25 | 33 | 48 | 57 | 63 | 76 | 82 | 88 | 95 |

1. How many comparisons are required to search for 57 using sequential search ?

5

1. How many comparisons are required to search for 35 using sequential search ?

4 until 48

1. How many comparisons are required to search for 57 using binary search ?

4

1. How many comparisons are required to search for 35 using binary search ?

Mid index = 4 (57>35), look left

Next Mid Index = (0+3)/2 = 1, 25 < 35, no -> look right  
 Mid Index = (2+3)/2 = 2, (33>35, look right)  
 Mid Index = (3+3)/2 = 3, (48 > 35, look left)

Left index = 4, right index = 3, stop. Not found

4 comparisons

1. Write a ***recursive*** sequential search function to search for a target in a sorted array of integer numbers. The function header is given below.

int search (int dataArray[], int arraySize, int start, int target)

int search(int dataArray[], int arraySize, int start, int target) {

    // Base case: if we have reached the end of the array or the size limit

    if (start >= arraySize) {

        return -1; // Target not found

    }

    // Check if the current element matches the target

    if (dataArray[start] == target) {

        return start; // Return the index of the target

    }

    // Recursive case: check the next element

    return search(dataArray, arraySize, start + 1, target);

}

1. Write a ***recursive*** binary search function to search for a target in a sorted array of integer numbers. The function header is given below.

int binarySearch (int dataArray[], int first, int last, int target)

int search(int dataArray[], int arraySize, int start, int target) {

    // Base case: if we have reached the end of the array or the size limit

    if (start >= arraySize) {

        return -1; // Target not found

    }

    // Check if the current element matches the target

    if (dataArray[start] == target) {

        return start; // Return the index of the target

    }

    // Recursive case: check the next element

    return search(dataArray, arraySize, start + 1, target);

}

1. Discuss and compare the time efficiency between **sequential search** and **binary search** algorithm.

Sequential Search:

Need to be linear, search from index 0 to n

Best case:

O(1)  
Avg case:

O(N)

Binary Search:

Best case O(1)

Worst case:

Log(n)

Area of search is halved per comparison