|  |  |
| --- | --- |
| **Data Structures & Algorithms**  Diploma in IT, ISF, FI  Year 2 (2018/19) Semester 4 | **Week 7** |
| **2 Hours** |
| **Tutorial 7 – Searching** | |

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*

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

*1*

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

*4*

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)

{

if (start < arraySize)

{

if(dataArray[start] == target)

return start;

else

{

if (dataArray[start] > target)

return -1;

else

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

}

}

else

return -1;

}

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

{

if (start > last)

return -1;

else

{

int mid = (start+last)/2;

if(dataArray[mid] == target)

return mid;

else

{

if (target < dataArray[mid])

return binarySearch(dataArray, start, mid-1, target);

else

return binarySearch(dataArray, mid+1, last, target);

}

}

}

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

*The binary search algorithm takes a shorter amount of time to search for the target as compared to the sequential search algorithm. This is because the binary search algorithm compares the middle index of the array and determines whether the target is in the first half or the second half of the array and searches only on the determined half of the array afterwards. This reduces the number of elements in the array to search by two-fold during each search.*