Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Report on '**Lab Work 1'** [COMP 314]

Submitted by:

Chandan Kumar Mahato (31)

III-year, II semester

Submitted to:

Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

Submission Date: Mar 9, 2023

Qlab1

1. ***Implement Linear Search, Binary Search Recursive, Binary Search Iterative algorithm and write tests.***

Solution:

Implementing Linear Search, Binary Search Recursive, Binary Search Iterative algorithm using JavaScript (node.js) and writing test using (jest) a JavaScript testing framework.

1. Linear Search Implementation:

```
function linearSearch(array, x) {
  for (i in array) if (array[i] == x) return i;
  return -1;
}
result = linearSearch([2, 4, 1, 3, 7, 0], 1);
result == -1
  ? console.log("Element Not Found")
  : console.log("Element found at index: ", result);
```

2. Recursive Binary Search Implementation:

```
function binarySearchRecursive(arr, low, high, x) {
  if (high >= low) {
    mid = Math.floor((high + low) / 2);
    if (arr[mid] == x) {
      return mid;
    } else if (arr[mid] > x) {
      return binarySearchRecursive(arr, low, mid - 1, x);
    } else return binarySearchRecursive(arr, mid + 1, high, x);
  } else return -1;
}
result = binarySearchRecursive([2, 3, 4, 10, 40], 0, 5, 10);
result == -1
  ? console.log("Element Not Found")
  : console.log("Element found at index: ", result);
```

3. Iterative Binary Search Implementation:

```
function binarySearchIterative(arr, x) {
  low = 0;
  high = arr.length - 1;
  mid = 0;
  while (low <= high) {
    mid = Math.floor((high + low) / 2);
    if (arr[mid] < x) {
      low = mid + 1;
    } else if (arr[mid] > x) {
      high = mid - 1;
    } else return mid;
  }
  return -1;
}
result = binarySearchIterative([2, 3, 4, 10, 40], 40);
result == -1
  ? console.log("Element Not Found")
  : console.log("Element found at index: ", result);

module.exports = { linearSearch, binarySearchRecursive, binarySearchIterative };
```

Test Impementation:

```
const {
binarySearchIterative,
binarySearchRecursive,l
inearSearch,} = require("../../index");
describe("Search Algo Test", () => {
  it("should return index of element using linear search", () => {
    const result = linearSearch([2, 4, 1, 3, 7, 0], 1);
    expect(result).toEqual("2");
  });
  it("should return index of element using binary search recursive", () => {
    const result = binarySearchRecursive([2, 3, 4, 10, 40], 0, 5, 10);
    expect(result).toEqual(3);
  });
  it("should return index of element using binary search iterative", () => {
    const result = binarySearchIterative([2, 3, 4, 10, 40], 40);
    expect(result).toEqual(4);
  });
});
```

Output:

```
default@LAPTOP-FLU6LLN1 MINGW64 /a/Chandan Semester Work/6th sem/Algorithm and complexity/Lab/DS
th_Sem/Lab1
$ node index.js
Element found at index:  2
Element found at index:  3
Element found at index:  4

default@LAPTOP-FLU6LLN1 MINGW64 /a/Chandan Semester Work/6th sem/Algorithm and complexity/Lab/DS
A_Lab_6th_Sem/Lab1
$ npm run test

> lab1_test_demo@1.0.0 test A:\Chandan Semester Work\6th sem\Algorithm and complexity\Lab\DSA_La
b_6th_Sem\Lab1
> jest --watchAll --verbose --coverage --detectOpenHandles

  console.log
    Element found at index:  2

      at Object.log (index.js:9:13)

  console.log
    Element found at index:  3

      at Object.log (index.js:24:13)

  console.log
    Element found at index:  4

      at Object.log (index.js:43:13)

PASS tests/unit/search.test.js
  Search Algo Test
    √ should return index of element using linear search (8 ms)
    √ should return index of element using binary search recursive (2 ms)
    √ should return index of element using binary search iterative (2 ms)

----------|---------|----------|---------|---------|-------------------
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------|---------|----------|---------|---------|-------------------
All files |   86.66 |    72.22 |     100 |   85.71 |
 index.js |   86.66 |    72.22 |     100 |   85.71 | 4,19,35,38
----------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        1.822 s
Ran all test suites.
```