IG Computer Science

Unit 1: Problem Solving

Part 8: Recursion

Topic 3: Sorting and Searching Algorithms

Lecture Contents

- Iterative Binary Search
- Recursive Binary Search
- Comparison of Iteration versus Recursion

Binary Search – Iterative

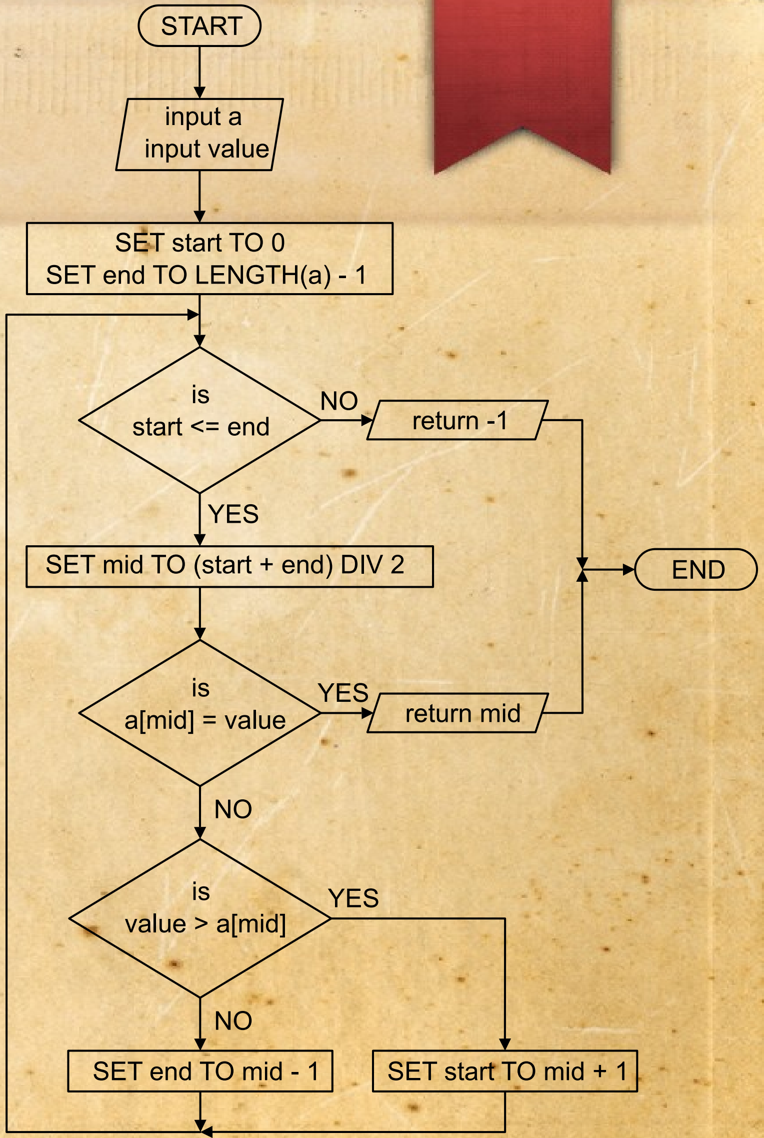
```
FUNCTION binarySearch(arr, value)
    SET start = 0
    SET end = LENGTH(arr) - 1

    WHILE start <= end DO
        SET mid TO (start + end) DIV 2
        IF arr[mid] = value THEN
            RETURN mid  // value found
        ELSE IF arr[mid] < value THEN
            SET start = end + 1 //Search in the right half
        ELSE
            SET end = mid - 1   //Search in the left half
        END IF
    END WHILE
    RETURN -1  // value not found
END FUNCTION
```

Binary Search – Iterative

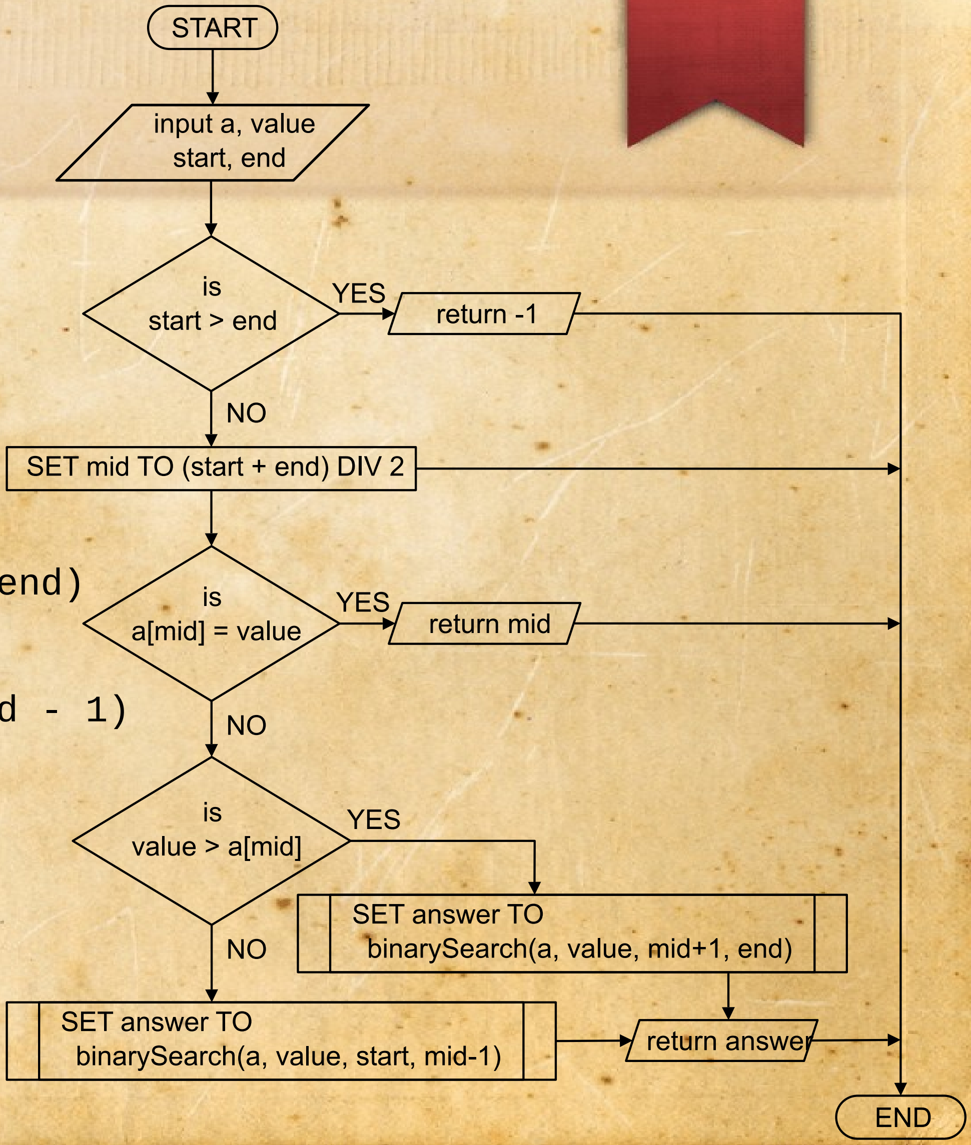
```
FUNCTION binarySearch(arr, value)
    SET start = 0
    SET end = LENGTH(arr) - 1

    WHILE start <= end DO // Iterative
        SET mid TO (start + end) DIV 2
        IF arr[mid] = value THEN
            RETURN mid  // value found
        ELSE IF arr[mid] < value THEN
            SET start = end + 1 //Search in the right half
        ELSE
            SET end = mid - 1   //Search in the left half
        END IF
    END WHILE
    RETURN -1  // value not found
END FUNCTION
```



Binary Search – Recursive

```
FUNCTION binarySearch(arr, value, start, end)
    IF start > end THEN
        RETURN -1  // value not found
    END IF
    SET mid = (start + end) DIV 2
    IF arr[mid] = value THEN
        RETURN mid  // value found
    ELSE IF value > arr[mid] THEN
        // Recursively search in the upper half
        RETURN binarySearch(arr, value, mid + 1, end)
    ELSE
        // Recursively search in the lower half
        RETURN binarySearch(arr, value, start, mid - 1)
    ENDIF
END FUNCTION
```



START

input a, value start, end

is start > end

YES

return -1

NO

SET mid TO (start + end) DIV 2

is a[mid] = value

YES

return mid

NO

is value > a[mid]

YES

SET answer TO binarySearch(a, value, mid+1, end)

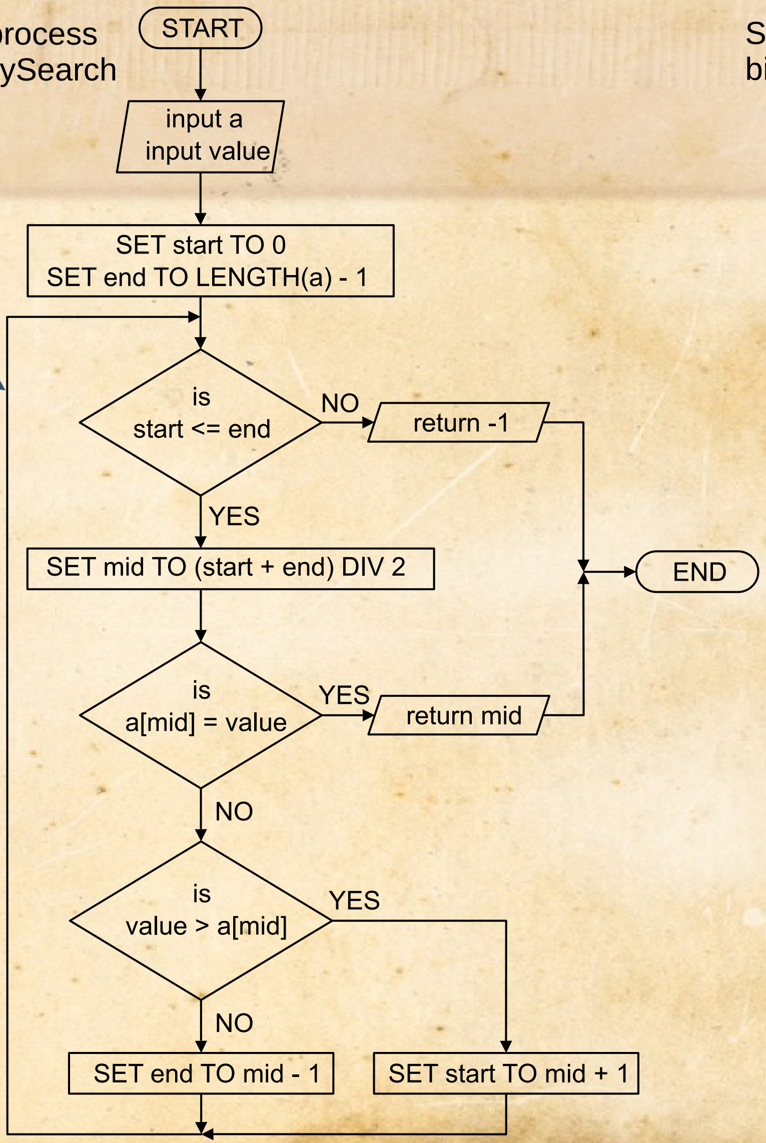
NO

SET answer TO binarySearch(a, value, start, mid-1)

return answer

END

Subprocess binarySearch



START

input a
input value

SET start TO 0
SET end TO LENGTH(a) - 1

is
start <= end

NO

return -1

YES

SET mid TO (start + end) DIV 2

is
a[mid] = value

YES

return mid

NO

is
value > a[mid]

YES

NO

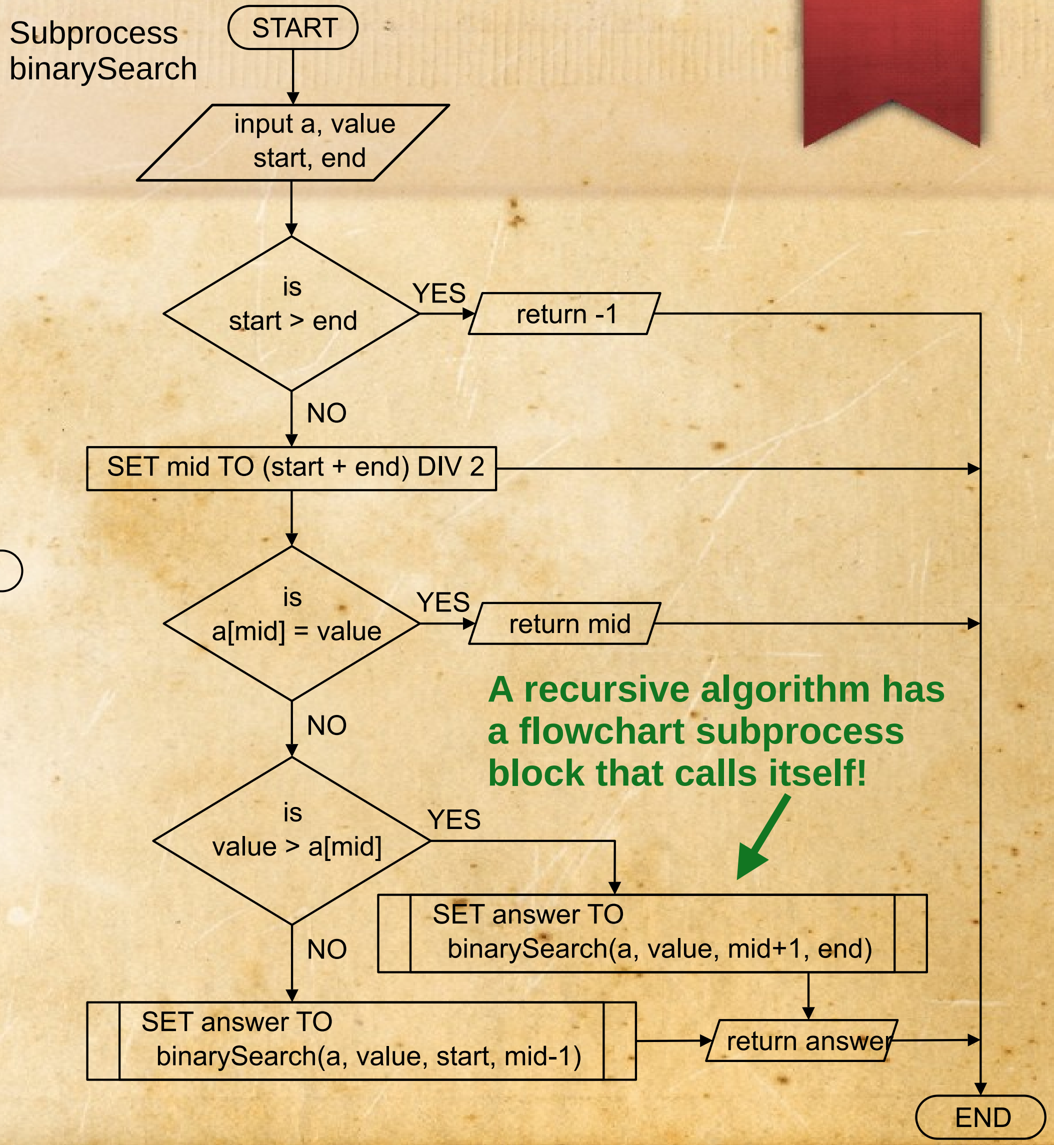
SET end TO mid - 1

SET start TO mid + 1

END

An iterative algorithm has a decision block that feeds back to an earlier point

Subprocess binarySearch



START

input a, value
start, end

is
start > end

YES

return -1

NO

SET mid TO (start + end) DIV 2

is
a[mid] = value

YES

return mid

NO

is
value > a[mid]

YES

SET answer TO
binarySearch(a, value, mid+1, end)

NO

SET answer TO
binarySearch(a, value, start, mid-1)

return answer

END

A recursive algorithm has a flowchart subprocess block that calls itself!

Binary Search

```
// Recursive
FUNCTION binarySearch(arr, value, start, end)
    IF start > end THEN
        RETURN -1  // value not found
    END IF

    SET mid = (start + end) DIV 2
    IF arr[mid] = value THEN
        RETURN mid  // value found
    ELSE IF value > arr[mid] THEN
        // Recursively search in the upper half
        RETURN binarySearch(arr, value, mid + 1, end)
    ELSE
        // Recursively search in the lower half
        RETURN binarySearch(arr, value, start, mid - 1)
    ENDIF
END FUNCTION
```

```
// Iterative
FUNCTION binarySearch(arr, value)
    SET start = 0
    SET end = LENGTH(arr) – 1

    WHILE start <= end DO  //ITERATION
        SET mid TO (start + end) DIV 2
        IF arr[mid] = value THEN
            RETURN mid  // value found
        ELSE IF arr[mid] < value THEN
            SET start = end + 1 //Search in the right half
        ELSE
            SET end = mid - 1   //Search in the left half
        END IF
    END WHILE
    RETURN -1  // value not found
END FUNCTION
```

Iteration versus Recursion

- Iterative algorithms have flowcharts with feedback to an earlier point in the algorithm
- Iterative algorithms use one of the loop key words (`REPEAT`, `WHILE`, or `FOR`)

- Recursive algorithms have flowcharts with a subprocess block that calls itself
- Recursive algorithms use functions that call themselves

IG Computer Science

Unit 1: Problem Solving

Part 8: Recursion

Topic 3: Sorting and Searching Algorithms