

# Recursive Binary Search

Lecture 35

Sections 9.1 - 9.2

Robb T. Koether

Hampden-Sydney College

Mon, Dec 9, 2013

1 A Recursive Sequential Search Function

2 A Recursive Binary Search Function

3 Assignment

# Outline

- 1 A Recursive Sequential Search Function
- 2 A Recursive Binary Search Function
- 3 Assignment

- A sequential search function can be written recursively.

# The sequentialSearch() Function

## The Recursive sequentialSearch() Function

```
int sequentialSearch(int arr[], int value, int left, int right)
{
    // See if the search has failed

    if (left > right)
        return -1;

    // See if the value has been found

    else if (arr[left] == value)
        return left;

    // Otherwise, continue the search

    else
        return sequentialSearch(arr, value, left + 1, right);
}
```

# The `binarySearch()` Function

- The signature of the sequential function is  
`(int arr[], int value, int left, int right)`
- This means that the initial function call would have to be  
`sequentialSearch(arr, value, 0, size - 1);`
- However, that is not the standard interface for a search function.

# The `binarySearch()` Function

- Normally, the function call would be written as

```
sequentialSearch(arr, size, value);
```

- Therefore, we should write an additional `binarySearch()` function with prototype

```
int sequentialSearch(int arr[], int size, int  
                    value);
```

- This function will call the other one and then report back the result.

# The sequentialSearch() Function

## The Non-Recursive sequentialSearch() Function

```
int sequentialSearch(int arr[], int size, int value)
{
    return sequentialSearch(arr, value, 0, size - 1);
}
```



# Outline

1 A Recursive Sequential Search Function

2 A Recursive Binary Search Function

3 Assignment

# The Binary Search Algorithm

- The binary search algorithm is naturally recursive.
- That is, the action that we perform on the original list is the very same as the action that we perform on the sublists.
- As a consequence, it is very easy to write a recursive binary search function.

# A Recursive Binary Search

- A binary search may also be written recursively.
- As before, we will write two versions.
- The “public” version with the standard interface.
- The “private” version with the recursive interface.

# The `binarySearch()` Function

## The Recursive `binarySearch()` Function

```
int binarySearch(int arr[], int value, int left, int right)
{
    // See if the search has failed

    if (left > right)
        return -1;

    // See if the value is in the first half

    int middle = (left + right)/2;

    if (value < arr[middle])
        return binarySearch(arr, value, left, middle - 1);

    // See if the value is in the second half

    else if (value > arr[middle])
        return binarySearch(arr, value, middle + 1, right);

    // The value has been found

    else
        return middle;
}
```

# The `binarySearch()` Function

- Normally, the function call would be written as

```
binarySearch(arr, size, value);
```

- Therefore, we should write an additional `binarySearch()` function with prototype

```
int binarySearch(int arr[], int size, int  
                value);
```

- This function will call the other one and then report back the result.

# The `binarySearch()` Function

## The Non-Recursive `binarySearch()` Function

```
int binarySearch(int arr[], int size, int value)
{
    return binarySearch(arr, value, 0, size - 1);
}
```

# Outline

- 1 A Recursive Sequential Search Function
- 2 A Recursive Binary Search Function
- 3 Assignment**

# Assignment

## Assignment

- Read Sections 9.1 - 9.2.