Recursive Binary Search

Lecture 36
Sections 9.1 - 9.2

Robb T. Koether

Hampden-Sydney College

Wed, Dec 5, 2018

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 `seqSearch()` Function

The Recursive `seqSearch()` Function

```
int seqSearch(int a[], 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 (a[left] == value)
        return left;

//  Otherwise, continue the search

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

The `binSearch()` Function

- The signature of the recursive function is

```
(int a[], int value, int left, int right)
```

- This means that the initial function call would have to be

```
seqSearch(a, value, 0, size - 1);
```

- However, that is not the standard interface for a search function.

The `binSearch()` Function

- Normally, the function call would be written as

```
seqSearch(a, size, value);
```

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

```
int seqSearch(int a[], int size, int value);
```

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

The seqSearch() Function

The Non-Recursive seqSearch() Function

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

- The non-recursive search function gets things started by passing the required parameters to the recursive search function.
- This is a common pattern with recursive functions.

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 binSearch() Function

The Recursive binSearch() Function

```
int binSearch(int a[], 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 < a[middle])
        return binSearch(a, value, left, middle - 1);

//  See if the value is in the second half

    else if (value > a[middle])
        return binSearch(a, value, middle + 1, right);

//  The value has been found

    else
        return middle;
}
```

The `binSearch()` Function

- Normally, the function call would be written as

```
binSearch(a, size, value);
```

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

```
int binSearch(int a[], int size, int value);
```

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

The binSearch() Function

The Non-Recursive binSearch() Function

```
int binSearch(int a[], int size, int value)
{
    return binSearch(a, 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.