CS302 - Data Structures using C++

Topic: Recursion with Arrays

Kostas Alexis



 Recursively displaying elements stored in an array



- Recursively displaying elements stored in an array
 - Start with first element



- Recursively displaying elements stored in an array
 - Start with first element

```
first

last

34 16 91 76 23 54

0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{

    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element

```
first

last

34

16

91

76

23

54

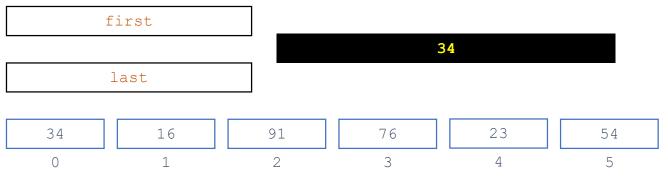
0

1

54

54
```

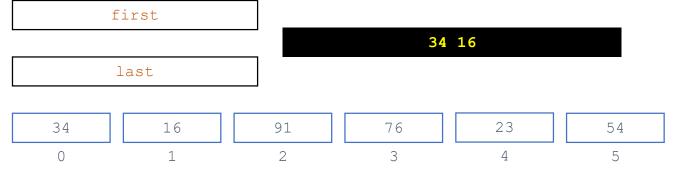
- Recursively displaying elements stored in an array
 - Start with first element



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
This function has an
```

implicit base case

- Recursively displaying elements stored in an array
 - Start with first element



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element

```
    first

    34 16 91

    34
    16
    91
    76
    23
    54

    0
    1
    2
    3
    4
    5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element

```
first

34 16 91 76

last

34 16 91 76

23 54

0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element

```
    first

    34 16 91 76 23

    34
    16
    91
    76
    23
    54

    0
    1
    2
    3
    4
    5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element

```
    first

    34 16 91 76 23 54

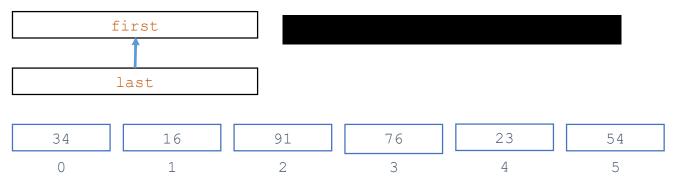
    1ast

    34
    16
    91
    76
    23
    54

    0
    1
    2
    3
    4
    5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```



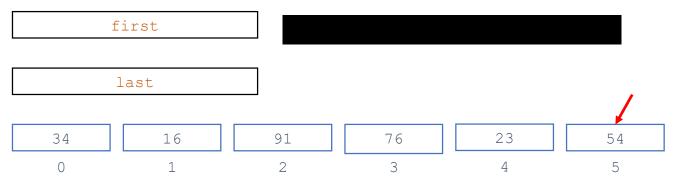
- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

```
first
        last
                                                       23
34
                           91
             16
                                         76
                                                                     54
0
                            2
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << myArray[first] << " ";</pre>
     if (first < last)</pre>
          displayArray(myArray, first + 1, last);
} // end displayArray
// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
     if (first <= last)</pre>
          displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
```



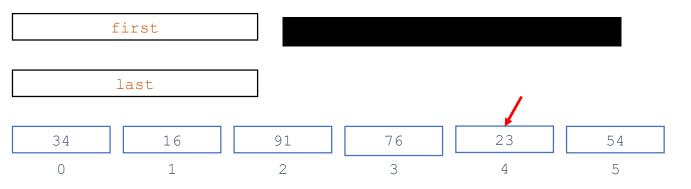
- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << myArray[first] << " ";</pre>
     if (first < last)</pre>
          displayArray(myArray, first + 1, last);
} // end displayArray
// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
     if (first <= last)</pre>
     displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

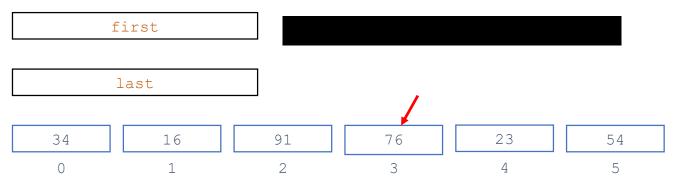


```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```



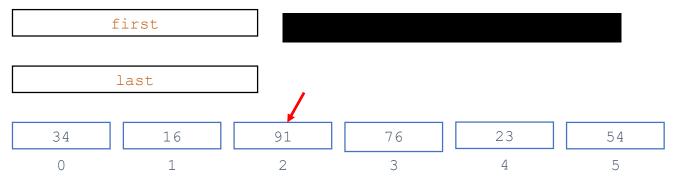
- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << myArray[first] << " ";</pre>
     if (first < last)</pre>
          displayArray(myArray, first + 1, last);
} // end displayArray
// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
     if (first <= last)</pre>
     displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

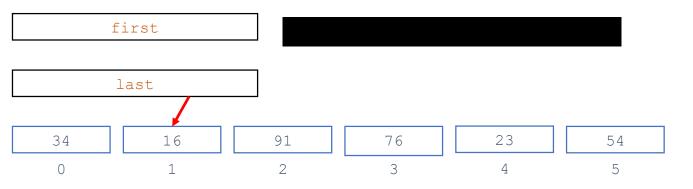


```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```



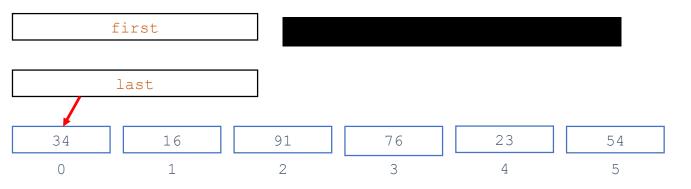
- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << myArray[first] << " ";</pre>
     if (first < last)</pre>
          displayArray(myArray, first + 1, last);
} // end displayArray
// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
     if (first <= last)</pre>
     displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

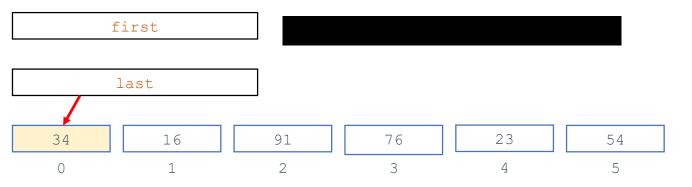


```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << myArray[first] << " ";</pre>
     if (first < last)</pre>
          displayArray(myArray, first + 1, last);
} // end displayArray
// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
     if (first <= last)</pre>
     displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
```

} // end displayArray



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element



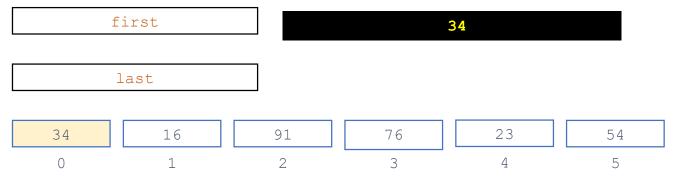
```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```

Base case reached



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element





- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

```
    first

    34 16

    34
    16
    91
    76
    23
    54

    0
    1
    2
    3
    4
    5
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

```
    first

    34 16 91

    1 ast

    34 16 91 76 23 54

    0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

```
    first

    34 16 91 76

    34 16 91 76

    34 16 91 76
    23 54

    0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
        } // end if
} // end displayArray</pre>
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

```
    first

    34 16 91 76 23

    1ast

    34 16 91 76 23 54

    0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element

```
    first

    34 16 91 76 23 54

    34
    16
    91
    76
    23
    54

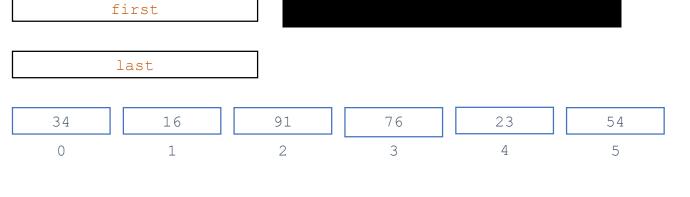
    0
    1
    2
    3
    4
    5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)
    {
        displayArray(myArray, first, last-1);
        std::cout << myArray[last] << " ";
    } // end if
} // end displayArray</pre>
```



- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << myArray[first] << " ";</pre>
     if (first < last)</pre>
          displayArray(myArray, first + 1, last);
} // end displayArray
// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
     if (first <= last)</pre>
          displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
// Divide the array in half
void displayArray(int myArray[], int first, int last)
     if (first == last)
          std::cout <<myArray[first] << " ";</pre>
     else
          int mid = (first + last)/2;
          displayArray(myArray, first, mid);
          displayArray(myArray, mid + 1, last);
     } // end if
} // end displayArray
```

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

```
first

last

34 16 91 76 23 54

0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

void displayArray(int myArray[], int first, int last)

// Start with myArray[last]

```
if (first <= last)</pre>
          displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
// Divide the array in half
void displayArray(int myArray[], int first, int last)
     if (first == last)
          std::cout <<myArray[first] << " ";</pre>
     else
          int mid = (first + last)/2;
          displayArray(myArray, first, mid);
          displayArray(myArray, mid + 1, last);
     } // end if
} // end displayArray
```

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

```
last

34 16 91 76 23 54
0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]
void displayArray(int myArray[], int first, int last)
{
    if (first <= last)</pre>
```

displayArray(myArray, first, last-1);
std::cout << myArray[last] << " ";</pre>

} // end if

} // end displayArray

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

```
last

91 76 23 54
0 1 2 3 4 5
```

```
// Start with myArray[first]
          void displayArray(int myArray[], int first, int last)
               std::cout << myArray[first] << " ";</pre>
               if (first < last)</pre>
                     displayArray(myArray, first + 1, last);
          } // end displayArray
          // Start with myArray[last]
          void displayArray(int myArray[], int first, int last)
               if (first <= last)</pre>
                     displayArray(myArray, first, last-1);
                     std::cout << myArray[last] << " ";</pre>
               } // end if
           } // end displayArray
          // Divide the array in half
          void displayArray(int myArray[], int first, int last)
               if (first == last)
                     std::cout <<myArray[first] << " ";</pre>
               else
                     int mid = (first + last)/2;
                 → displayArray (myArray, first, mid);
two calls
                 → displayArray (myArray, mid + 1, last);
```

} // end if
} // end displayArray

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

```
last

10 91 76 23 54
0 1 2 3 4 5
```

```
// Start with myArray[first]
          void displayArray(int myArray[], int first, int last)
               std::cout << myArray[first] << " ";</pre>
               if (first < last)</pre>
                     displayArray(myArray, first + 1, last);
          } // end displayArray
          // Start with myArray[last]
          void displayArray(int myArray[], int first, int last)
               if (first <= last)</pre>
                     displayArray(myArray, first, last-1);
                     std::cout << myArray[last] << " ";</pre>
               } // end if
           } // end displayArray
          // Divide the array in half
          void displayArray(int myArray[], int first, int last)
               if (first == last)
                     std::cout <<myArray[first] << " ";</pre>
               else
                     int mid = (first + last)/2;
                 → displayArray (myArray, first, mid);
two calls
                 → displayArray (myArray, mid + 1, last);
                } // end if
```

} // end displayArray

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

```
last

91 76 23 54
0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray

// Start with myArray[last]</pre>
```

} // end displayArray

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

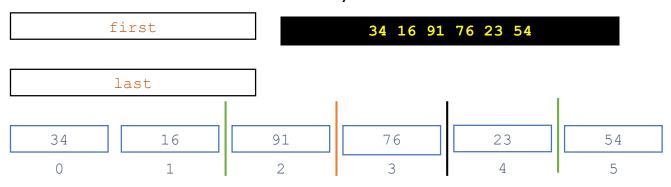
```
first
        last
34
                           91
                                                       23
                                                                    54
             16
                                         76
0
```

```
// Start with myArray[first]
          void displayArray(int myArray[], int first, int last)
               std::cout << myArray[first] << " ";</pre>
               if (first < last)</pre>
                     displayArray(myArray, first + 1, last);
          } // end displayArray
          // Start with myArray[last]
          void displayArray(int myArray[], int first, int last)
               if (first <= last)</pre>
                     displayArray(myArray, first, last-1);
                     std::cout << myArray[last] << " ";</pre>
                } // end if
           } // end displayArray
          // Divide the array in half
          void displayArray(int myArray[], int first, int last)
               if (first == last)
                     std::cout <<myArray[first] << " ";</pre>
                else
                     int mid = (first + last)/2;
                 → displayArray(myArray, first, mid);
two calls
```

→ displayArray (myArray, mid + 1, last);

} // end if } // end displayArray

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half



```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
{
    std::cout << myArray[first] << " ";
    if (first < last)
        displayArray(myArray, first + 1, last);
} // end displayArray</pre>
```

void displayArray(int myArray[], int first, int last)

// Start with myArray[last]

```
if (first <= last)</pre>
          displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
// Divide the array in half
void displayArray(int myArray[], int first, int last)
     if (first == last)
          std::cout <<myArray[first] << " ";</pre>
     else
          int mid = (first + last)/2;
          displayArray(myArray, first, mid);
          displayArray(myArray, mid + 1, last);
     } // end if
} // end displayArray
```

- Recursively displaying elements stored in an array
 - Start with first element
 - Start with last element
 - Divide the array in half

```
    first

    34 16 91 76 23 54

    34 16 91 76 23 54

    0 1 2 3 4 5
```

```
// Start with myArray[first]
void displayArray(int myArray[], int first, int last)
     std::cout << mvArray[first] << " ";</pre>
    if (first < last)</pre>
          displayArray(myArray, first + 1, last);
// end_displayArray
         Implicit base case
// Start with myArray last
void displayArray(int myArray[], int first, int last)
    if (first <= last)</pre>
          displayArray(myArray, first, last-1);
          std::cout << myArray[last] << " ";</pre>
     } // end if
} // end displayArray
// Divide the array in half
void displayArray(int myArray[], int first, int last)
                             explicit base case
     if (first == last)
          std::cout <<myArray[first] << " ";</pre>
     else
          int mid = (first + last)/2;
          displayArray(myArray, first, mid);
          displayArray(myArray, mid + 1, last);
     } // end if
} // end displayArray
```

Thank you

