More Arrays

Last Time

- We
  - Learned how to pass arrays to functions
- Today we will
  - Start talking about searching arrays

Searching Arrays

- We search an array to find a particular element in an array.
- For example, we might like to search an array of student grades for all students who got higher than 90% (i.e. A’s).
- How would we do this?

Sequential or Linear Search

- Compare each element of the array with the value (or key) that we are searching for.
- This is called linear or sequential search.
- Linear Search Algorithm:
  - For each array element
    - If the current element contains the target
      - Return the subscript of the current element
    - Return -1

Write the function findElement

```cpp
int findElement(int [], int i);
int main()
{
  int grades[10];
  int element, index = -1;
  for(int i = 0; i < 10; i++)
    cin >> grades[i];
  cout << "Which element would you like to find?" << endl;
  cin >> element;
  findElement(grades, element, index);
  if(-1 == index)
    cout << "Element could not be found!" << endl;
  else
    cout << "Element was found at index " << index << endl;
}
```

Function to find element

```cpp
void findElement( int ar[],
                 int x,
                 int & index)
{
  for(int i = 0; i < 10; i++)
    if(ar[i] == x)
      index = i;
}
```
Problem

- 20.1: Write a function to return the index of the smallest element in a subarray.
- A subarray is a section of an array. The subarray is determined by its starting and ending indexes.
- The function will have the following arguments:
  - The array,
  - The starting index of the subarray,
  - The ending index of the subarray,
  - The index of the smallest element.

Function findIndexOfMin

```cpp
void findIndexOfMin(const int x[], int startIndex, int endIndex, int& index)
{
    You fill in the rest
}
```

Summary

- In today’s lecture we covered
  - Searching arrays
- Readings
  - Chapter 4