COMP 110

Introduction to Search Algorithms

Darrell Bethea
June 8, 2011
Questions?
Today in COMP 110

- Search Algorithms
The array itself is referred to by the name `scores` (in this particular case).

<table>
<thead>
<tr>
<th>Indices</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>68</td>
<td>73</td>
<td>57</td>
<td>102</td>
<td>94</td>
</tr>
</tbody>
</table>

The array scores: `scores[3]`
System.out.println("Enter 5 basketball scores:");

int[] scores = new int[5];
int scoreSum = 0;
for (int i = 0; i < scores.length; i++)
{
    scores[i] = keyboard.nextInt();
    scoreSum += scores[i];
}
double average = (double) scoreSum / scores.length;
System.out.println("Average score: "+ average);

for (int i = 0; i < scores.length; i++)
{
    if (scores[i] > average)
        System.out.println(scores[i] + ": above average");
    else if (scores[i] < average)
        System.out.println(scores[i] + ": below average");
    else
        System.out.println(scores[i] + ": equal to the average");
}
Introduction to sorting

- Given an array of numbers, sort the numbers into ascending order

- Input array:

  4  7  3  9  6  2  8

- Sorted array:

  2  3  4  6  7  8  9
public static void printArray(int[] array)
{
    for(int i = 0; i < array.length; i++)
        System.out.print(array[i] + " ");
    System.out.println();
}
Searching an Array

- Write an algorithm to search for an item in an unsorted list
  - Start with finding the index of 5 in the list
    - 3 6 2 1 7 8 5 9 10 3
public int findIndex(int x, int[] array) {
    for(int i =0; i < array.length; i++) {
        if(array[i] == x) {
            return i;
        }
    }
    return -1;
}
What if the array is sorted?

- Do we need to start at the beginning and look all the way through?
- What if we start in the middle of the array?
  - int midPoint = (array.length / 2);
- Since the array is sorted:
  - If array[midPoint] > x, we know x must be in first half of array
  - Else if array[midPoint] > x, x must be in 2nd half of array
  - Else if array[midPoint] == x, return midPoint
- This is called a binary search
public int binarySearch(int x, int[] array) {
    int min = 0;
    int max = array.length-1;
    while (min <= max) {
        int mid = (min + max) / 2;
        if (array[mid] > x)
            max = mid - 1;
        else if (array[mid] < x)
            min = mid + 1;
        else // Must be equal
            return mid;
    }
    return -1;
}