Design File

Assignment 1 CSC130

Derek Irvin

February 23rd, 2014

Problem: Write a program to find the nth largest number in two sorted arrays.

User Input: Two Sorted Arrays

Method Divide and Conquer

Divide and Conquer is based on breaking a big problem into sub problems and solving the sub problems to get the answer.

Here we have two sorted (already in order) arrays and we are ask to find the nth largest number between them.

Since they are already sorted assuming the array size of A and B >= to N. We only need to look at the first N values of the integers because any more and the value would be higher than the Nth value of these Sorted Arrays.

Now using Divide and Conquer we break the problem down in smaller problems to solve. Now that we only need to look at the first N values of the array we break the problem down even more dividing the size of the arrays in half and comparing the mid values of each. After comparing the values we can divide the problem down more. If A[M] > B[M] this means that any value from A[m + 1] to A[N] would be too big at the median to be the Nth value leaving us with A[0] to A[M]. For B[M] Any Value from B[0] to B[M-1] is too small since B[m] was less than that of the median value of A[] leaving us with B[m] - B[K] as a possible value.

From here we repeat dividing and comparing middle values until we have 3 values left leaving the median as the Nth Greatest Value.

PseudoCode:

Declarations:

Declare Arrays - Set to a high enough value for a user’s choice.

Declare Nth Value we will look for.

Prompt the user for the number of values in the Array.

-user input

Confirm user value is greater than N Else there isn't enough values to find the Nth Value.

User Message if Array Size < N and End Program

-Take user input for array values

Find the mid value of the two arrays by dividing N / 2.

Compare Mid values of A[] and B[]

if A[M] > B[M]

A[] = A[0] - A[M]

B[] = B[M] - B[N]

else

A[] = A[M] - A[N]

B[] = B[0] - B[M]

end if

Loop:

Compare number of elements

If Greater than 3 else go to greatestValue

Mid Value = ArraySize/2

Compare A[M] to B[M]

if A[M] > B[M]

A[] = A[0] - A[M]

B[] = A[M] - A[N]

else

A[] = A[M] - A[N]

B[] = B[0] - B[M]

end if

Return to Loop

greatestValue

compare values of A[] and B[] to find median

Display Median as the Nth Greatest Number

End Program.

import java.util.Scanner;  
import java.util.\*;  
import java.lang.\*;  
   
class GetInputFromUser  
{  
 public static void main(String args[])  
 {  
 int arraySize; // Size of the Array  
   
 Scanner in = new Scanner(System.in);  
   
 // get the size of the array  
 System.out.println("what is the size of your arrays?");  
 arraySize = in.nextInt();  
 System.out.println("You entered integer "+arraySize);  
   
 // Filling the arrays  
 int[] arrayA = new int[arraySize];  
   
   
 System.out.println("Please fill the first array: ");  
 for(int j=0; j<arraySize; j++)  
 arrayA[j]=in.nextInt();  
   
// Testing that the elements are filling the array  
// System.out.println("Array A elements are: ");  
// for(int i=0; i<arraySize; i++)  
// System.out.println(arrayA[i]+",");  
   
 int[] arrayB = new int[arraySize];  
   
 System.out.println("Please fill the second array: ");  
 for(int j=0; j<arraySize; j++)  
 arrayB[j]=in.nextInt();  
   
// testing that the elements are filling the array  
// System.out.println("Array B elements are: ");  
// for(int i=0; i<arraySize; i++)  
// System.out.println(arrayB[i]+",");  
 int middleValue = arraySize/2;  
 // for divide and conquer, base case middle value  
   
 int middleArrayA = arrayA[middleValue];  
 int middleArrayB = arrayB[middleValue];  
   
 int[] dividedArrayA = new int[middleValue];  
 int[] dividedArrayB = new int[middleValue];  
  
   
  
   
 while(arrayA.length + arrayB.length > 3)  
 {  
 if(middleArrayA > middleArrayB)  
 {  
 // if the middle value of A is greater then that of the middle value of B we know that the value cannot be   
 // in the higher half of A and it cannot be in the lower half of B.   
 System.out.println("The Middle of A is Greater");  
 // So we will want to copy the lower half of Array A   
 // Stystem.arrayCopy(Source: arrayA  
 System.arraycopy(arrayA, 0, dividedArrayA, 0, middleValue);  
 System.arraycopy(arrayB, middleValue, dividedArrayB, 0, middleValue);  
 for (int i = 0; i < middleValue; i++)  
 System.out.println(dividedArrayA[i]);  
 for (int i = 0; i < middleValue; i++)   
 System.out.println(dividedArrayB[i]);  
 arraySize = middleValue;  
 }  
 else  
 {  
 System.out.println("The Center of array B is Greater");  
 System.arraycopy(arrayA, middleValue, dividedArrayA, 0, middleValue);  
 System.arraycopy(arrayB, 0, dividedArrayB, 0, middleValue);  
 for (int i = 0; i < middleValue; i++)  
 System.out.println(dividedArrayA[i]);   
 for (int i = 0; i < middleValue; i++)  
 System.out.println(dividedArrayB[i]);  
 arraySize = middleValue;  
 }  
 }  
 }  
}