

Program 1

Due: 5:00pm, February 6, 2009

For this programming assignment you will implement and test two different C++ solutions to the *IndexEqual* problem you worked on in Homework 1. The details follow.

1. Implement the efficient recursive version of the **IndexEqual(A, i, j)** algorithm described in Homework 1.
2. Implement an iterative version of **IndexEqual(A, i, j)** that merely loops through each of the elements from **A[i]** to **A[j]** looking for an entry such that **A[k]=k**. Your implementation should exit the loop upon finding such an element.
3. As an estimate of the running time, you should record the number of comparisons between **A[k]** and **k** performed in each algorithm, where a comparison is any of **==, !=, <, >, <=** or **>=**. You may use a global variable to do this.
4. To test your two algorithms you will try 100 different arrays, each of size 100 (actually the same array, but with different contents each time). The *k*th array will set **A[k] = k** and set the remaining elements to values not equal to their index. Remember that all the elements of the array must be unique integers and in order. You can store negative integers in the array. For each array you should output the number of the array (*k*) and the number of comparisons performed by each of the two algorithms on that array. At the end, you should also output the total number of comparisons performed by each algorithm over all 100 arrays.
5. Create a **readme.txt** file that describes exactly how to compile and execute your program and on what platform.
6. Collect your source code, readme file and any other files needed to compile and execute your program into one ZIP file called **<your_last_name>-pgm1.zip** and include it as an attachment to an email to me (holder@wsu.edu) by the deadline. Grading will be based not only on correctness, but also on programming style and documentation.