

Washington State University  
School of Electrical Engineering and Computer Science  
Spring 2022

CptS 223 Advanced Data Structures in C++

**Homework 2**

Due: January 26, 2022 (11:59pm pacific time)

**General Instructions:** Create a zip file containing all your files for this homework. Upload the zip file as your submission for Homework 2 for the course CptS 223 Pullman on the Canvas system by the above deadline.

For this homework 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. This function should be initially called with **i=0** and **j=99**.
2. Implement an iterative version of **IndexEqual** (**A**, **i**, **j**) that 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. This function should always be called with **i=0** and **j=99**.
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. Your **main** function should evaluate your two algorithms using 100 different arrays, each of size 100. 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 sorted in increasing 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. See next page for sample output.
5. Create a **readme.txt** file that describes exactly how to compile and execute your program. A **Makefile** is recommended, but optional. For this homework a **.h** file is also optional. We will be compiling and running the programs on the Ubuntu Linux platform using the Gnu C++ compiler. Your code must compile and run for us on this platform.
6. Collect your source code, readme file and any other files needed to compile and execute your program into one ZIP file and submit on Canvas. Grading will be based not only on correctness, but also on programming style and documentation. See the course syllabus for links to style guides.

Sample Output:

k	Recursive	Iterative
-----	-----	-----
0	11	1
1	13	2
2	9	3
3	11	4
4	13	5
5	7	6
6	11	7
7	13	8
8	9	9
9	11	10
10	13	11
11	5	12
12	11	13
...		
79	13	80
80	7	81
81	11	82
82	13	83
83	9	84
84	13	85
85	11	86
86	13	87
87	5	88
88	11	89
89	13	90
90	9	91
91	11	92
92	13	93
93	7	94
94	11	95
95	13	96
96	9	97
97	13	98
98	11	99
99	13	100
-----	-----	-----
Total	1060	5050