

CSC 225 FALL 2024  
ALGORITHMS AND DATA STRUCTURES I  
ASSIGNMENT 3 - PROGRAMMING  
UNIVERSITY OF VICTORIA

Complete the *match* function using JAVA or C++. Submit source code in **ArrayMatch.java** or **ArrayMatch.cpp**. Your code determines if a recursive *match* (something we are defining for this particular problem) can be found when examining two arrays, **A** and **B**. **A** and **B** are arrays of size **n**, containing the same number of **integer** elements.

Two arrays, **A** and **B**, are defined to be recursively *matches* of one another if at least one of the following two conditions is satisfied:

- I. **A** = **B** (the arrays have the same elements at each index)
- II. If **n** is divisible by 2, **A** and **B** are divided into two sub-arrays of equal size (**A** is divided into **A<sub>1</sub>** and **A<sub>2</sub>**, **B** into **B<sub>1</sub>** and **B<sub>2</sub>**). Then, at least one of the following conditions is satisfied:
  - a) (**A<sub>1</sub>** matches **B<sub>1</sub>**) AND (**A<sub>2</sub>** matches **B<sub>2</sub>**)
  - b) (**A<sub>1</sub>** matches **B<sub>1</sub>**) AND (**A<sub>1</sub>** matches **B<sub>2</sub>**)
  - c) (**A<sub>2</sub>** matches **B<sub>1</sub>**) AND (**A<sub>2</sub>** matches **B<sub>2</sub>**)

**Notes:**

1. If **n** is not divisible by 2, condition II will be not satisfied directly.
2. The match is recursively. If one split cannot find the match and sub-array's **n** is still divisible by 2, then you should do further splits, until there is only one item in array if necessary.
3. Hint: You can use the idea from Quick-sort or Merge-sort.
4. You **cannot** change the method signature for *match()* at all (two integer arrays as parameters, and returns a Boolean) or you will receive a score of **0**.
5. If your submission fails to compile, you will receive a score of **0**.
6. Submit the source code! If you submit the executable code (which is not necessary to submit) only, you will receive a score of **0**.
7. The course code and input files will be placed in same directory. The code should read one input file each time to execute.
8. You are welcome to create additional methods to aid in your implementation, but again, the *match()* method must return a Boolean when given two integer arrays.
9. Your course code will be compiled and executed in Linux.csc.uvic.ca remote server with following command lines:
  - JAVA:
    - javac ArrayMatch.java
    - java ArrayMatch input.txt
  - Or in C++:
    - g++ ArrayMatch.cpp -o ArrayMatch
    - ./ArrayMatch input.txt
  - You can use any IDE to do the coding. **But check your work on the server before you submit it.**
10. If you registered in this course, you should be able to login the remote server with command line: **"ssh your\_uvic\_email\_prefix@linux.csc.uvic.ca"** with the same password you login your UVic online account. Try it and check if you can login as soon as possible, if you cannot login you should talk to CSC department.
11. After login, **DO NOT** try "sudo", just run your code as a normal user.

**Input Format**

Input files must be three lines, formatted in the following way:

<a single integer representing the size of the arrays>

<integer elements for Array A, where the elements are separated by white space>

<integer elements for Array B, where the elements are separated by white space>

You have been provided with some sample files. It is *strongly* recommended you add further tests.

The first line has an integer  $1 \leq n \leq 10^4$ . The second line has the  $n$  integers,  $a_1, a_2, \dots, a_{n-1}$ , of array  $A$ , and the third line has the  $n$  integers,  $b_1, b_2, \dots, b_{n-1}$ , of array  $B$ , where  $0 \leq a_i, b_i \leq 10^8$ , for each  $0 \leq i \leq n - 1$ .

On one line print "YES" **in terminal**, if the arrays match, and "NO" if they do not. This output is case-sensitive, and the quotes are just for clarity.

### Sample Input 0

```
5
10 11 8 19 11
10 11 8 19 11
```

### Sample Output 0

```
YES
```

### Explanation 0

In this sample, condition 1 is satisfied since the two arrays are the same.

### Sample Input 1

```
8
10 2 8 9 3 7 4 1
10 2 8 9 4 1 4 1
```

### Sample Output 1

```
YES
```

### Explanation 1

In this sample, condition 2 is satisfied in a recursive manner. In fact,  $A_1$  and  $B_1$  are the same and so they are a match, as well. Moreover,  $A_2$  and  $B_2$  are a match recursively, by condition II.(c):

$$A_{2-2} = [4,1] = B_{2-1} = [4,1] = B_{2-2} = [4,1]$$

### Sample Input 2

```
6
10 2 8 9 3 7
10 10 10 9 3 7
```

### Sample Output 2

```
NO
```

### Explanation 2

The arrays are not exactly the same so condition 1 is not satisfied. To check other conditions, we divide the arrays in half but none of the conditions are satisfied, even recursively.

## Submission

You must solve this problem using Java or C++. You will read from standard input and print to standard output.

## Evaluation Criteria

The programming assignment will be marked out of 20. The following score ranges will apply to this assignment; to reach top of the ranges you will need to include a correct analysis if the worst-case runtime of your algorithm (you can type the analysis in your code as a comment, or you can also submit the analysis as a separate file. If your analysis is a separate file, it must be .pdf !):

Score	Description
0	Submission does not compile.
4	Compiles, but incorrectly reports YES or NO.
8	Compiles and correctly reports YES or NO, but no worst-case runtime analysis.
12	Correctly reports the solution but does so in $T(n) \in O(n^{2+\varepsilon})$ time, where $\varepsilon > 0$ .
16	Correctly reports the solution but does so in $T(n) \in O(n^2)$ time.
20	Correctly reports the solution in $T(n) \in O(n^{2-\varepsilon})$ time, where $\varepsilon > 0$ .