Problem Set 3 Exercise #14: Set Containment

Reference: Lecture 6 notes

Learning objectives: One-dimensional array; Algorithm design

Estimated completion time: 50 minutes

Problem statement:

Write a program **PS3_Ex14_SetContainment.java** to read two arrays **`arrA`** and **`arrB`** of distinct **`int`** values and check whether numbers in **`arrA`** is a subset of numbers in **`arrB`**.

For example, array {1, 3, 5} is a subset of another array {1, 5, 3, 9}**.**

Your program should contain a static method

```
int[] scanArray(Scanner sc)
```

that reads elements into an array and return it back to the **`main()`**;

and another static method

```
boolean isSubset(int[] arrA, int[] arrB)
```

that checks if numbers in **`arrA`** is a subset of numbers in **`arrB`**; it returns true if so, or false otherwise.

Note:

1. This task is quite tough. It requires both universal (for all) and existential (there exists) argument, and involves nested loops. Please think about your algorithm carefully before coding. A flawed/complicated algorithm makes your coding and debugging much tougher.
2. For a program to work on CodeCrunch, it must not have more than one Scanner object per program. Therefore in the skeleton program, we create a Scanner object in the **`main()`** method and pass it to **`scanArray()`** method.

Sample run #1:

```
Size of 1st array? 4
Enter 4 values: 14 5 1 9
Size of 2nd array? 7
Enter 7 values: 2 9 3 14 5 6 1
1st array is a subset of 2nd array
```

Sample run #2:

```
Size of 1st array? 4
Enter 4 values: 14 5 1 9
Size of 2nd array? 6
Enter 6 values: 2 3 14 5 6 1
1st array is not a subset of 2nd array
```