Set Abstract Data Type

In computer science, a **set** is an abstract data type that can store unique values. Write a Java implementation of Set ADT to store integers. Follow the specifications given below.

Class Specification

1. Name the class as Set
2. Have a default constructor to initialize the set

List of Public Methods

1. int size() - Marks - 0.25
2. boolean contains(int) - Marks - 0.25
3. String toString() - Marks - 0.25
4. void add(int) Marks - 0.25
5. void add(int[]) Marks - 0.50
6. Set intersection(Set) Marks - 1
7. Set retainAll(int[]) Marks - 1.5
8. int[][] cartesianProduct(Set) Marks - 2

Public Method Specification

int size()

Return the number of items in the set.

boolean contains(int)

Return true if the given item is present in the set, false otherwise.

String toString()

Return a string that represents the items in the set. For example:
A set having the integers 1, 2, 3, 4, 5 should return the string {1, 2, 3, 4, 5}

void add(int)

Add the given integer to the Set if it is not already present.
Caution: ADT client may add arbitrary number of integers. So, resize the array when full.

void add(int[])

Add the items given in the int array to the Set if they are not already present.

Set intersection(Set)

Return a Set that contains common elements of this set and the given set. The order of the elements in the new Set should match the order of the elements in this set. For example:
this - {1, 5, 2, 4, 3}
that - {4, 1, 3}
intersection - {1, 4, 3}

Set retainAll(int[])

Return a Set with the items that are contained in the given int array. For example:
this - {1, 5, 2, 4, 3}
int[] - [4, 1, 3]
result - {1, 4, 3}

int[][] cartesianProduct(Set)

Return a 2D array to represent the ordered pairs that results from the cartesian product of this Set to the given Set. For example:
this - {1, 5}
that - {4, 1, 3}
int[][] - [[1, 4], [1, 1], [1, 3], [5, 4], [5, 1], [5, 3]]
Caution: Return null if the cartesian product results in a empty set. For example:
this - {}
that - {1, 2, 3}
result - null