Create a Set Class

In this exercise we are going to create a class named `Set` to emulate an abstract data structure called "set". A set is like an array, but it cannot contain duplicate values. The typical use for a set is to simply check for the presence of an item. We can see how the ES6 `Set` object works in the example below:

```
const set1 = new Set([1, 2, 3, 5, 5, 2, 0]);
console.log(set1);
// output: {1, 2, 3, 5, 0}
console.log(set1.has(1));
// output: true
console.log(set1.has(6));
// output: false
```

First, we will create an add method that adds a value to our set collection as long as the value does not already exist in the set. Then we will create a remove method that removes a value from the set collection if it already exists. And finally, we will create a size method that returns the number of elements inside the set collection.

Create an `add` method that adds a unique value to the set collection and returns `true` if the value was successfully added and `false` otherwise.

Create a `remove` method that accepts a value and checks if it exists in the set. If it does, then this method should remove it from the set collection, and return `true`. Otherwise, it should return `false`. Create a `size` method that returns the size of the set collection.

Your `Set` class should have an `add` method.

	The add method should not add duplicate values.
	Your add method should return true when a value has been successfully added.
	Your add method should return false when a duplicate value is added.
	Your Set class should have a remove method.
	Your remove method should only remove items that are present in the set.
	Your remove method should remove the given item from the set.
	Your Set class should have a size method.
	The size method should return the number of elements in the collection.