Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // Best: O(n^2) time | O(1) space
  // Average: O(n^2) time | O(1) space
  // Worst: O(n^2) time | O(1) space
  public static int[] selectionSort(int[] array) {
    if (array.length == 0) {
      return new int[] {};
    }
    int startIdx = 0;
    while (startIdx < array.length - 1) {
      int smallestIdx = startIdx;
      for (int i = startIdx + 1; i < array.length; i++) {
        if (array[smallestIdx] > array[i]) {
          smallestIdx = i;
        }
      }
      swap(startIdx, smallestIdx, array);
      startIdx++;
    }
    return array;
  }

  public static void swap(int i, int j, int[] array) {
    int temp = array[j];
    array[j] = array[i];
    array[i] = temp;
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  public static int[] selectionSort(int[] array) {
    // Write your code here.
    return null;
  }
}

```

Custom Output Submit Code

Run or submit code when you're ready.

Our Tests

```
class ProgramTest {
  @Test
  public void TestExample() {
    int[] expected = {};
    int[] input = {};
    Utils.assertTrue(Compare(Program.solution(input), expected));
  }

  @Test
  public void TestExample() {
    int[] expected = {3, 6};
```