Our Solution(s)   Run Code

Solution 1   Solution 2

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

class Program {
  // O(n) time | O(n) space
  public static int maxSubsetSumNoAdjacent(int[] array) {
    if (array.length == 0) {
      return 0;
    } else if (array.length == 1) {
      return array[0];
    }
    int[] maxSums = array.clone();
    maxSums[1] = Math.max(array[0], array[1]);
    for (int i = 2; i < array.length; i++) {
      maxSums[i] = Math.max(maxSums[i - 1], maxSums[i - 2] + arr
    }
    return maxSums[array.length - 1];
  }
}
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```java
class Program {
  public static int maxSubsetSumNoAdjacent(int[] array) {
    // Write your code here.
    return -1;
  }
}
```

Our Tests   Custom Output   Submit Code

```
    @Test
    public void TestCase0() {
        int[] input = {};
        Utils.assertTrue(Program.maxConsecutiveNumbers(input) == 0);
    }

    @Test
    public void TestCase1() {
        int[] input = {1};
        Utils.assertTrue(Program.maxConsecutiveNumbers(input) == 1);
    }

    @Test
    public void TestCase2() {
        int[] input = {1, 2};
        Utils.assertTrue(Program.maxConsecutiveNumbers(input) == 2);
    }

    @Test
    public void TestCase3() {
        int[] input = {1, 2, 3};
```