```
1  /** Returns the sum of the integers in given array. */
2  public static int example1(int[ ] arr) {
3    int n = arr.length, total = 0;
4    for (int j=0; j < n; j++)                    // loop from 0 to n-1
5      total += arr[j];
6    return total;
7  }
8
9  /** Returns the sum of the integers with even index in given array. */
10 public static int example2(int[ ] arr) {
11   int n = arr.length, total = 0;
12   for (int j=0; j < n; j += 2)                 // note the increment of 2
13     total += arr[j];
14   return total;
15 }
16
17 /** Returns the sum of the prefix sums of given array. */
18 public static int example3(int[ ] arr) {
19   int n = arr.length, total = 0;
20   for (int j=0; j < n; j++)                    // loop from 0 to n-1
21     for (int k=0; k <= j; k++)                 // loop from 0 to j
22       total += arr[j];
23   return total;
24 }
25
26 /** Returns the sum of the prefix sums of given array. */
27 public static int example4(int[ ] arr) {
28   int n = arr.length, prefix = 0, total = 0;
29   for (int j=0; j < n; j++) {                  // loop from 0 to n-1
30     prefix += arr[j];
31     total += prefix;
32   }
33   return total;
34 }
35
36 /** Returns the number of times second array stores sum of prefix sums from first. */
37 public static int example5(int[ ] first, int[ ] second) { // assume equal-length arrays
38   int n = first.length, count = 0;
39   for (int i=0; i < n; i++) {                  // loop from 0 to n-1
40     int total = 0;
41     for (int j=0; j < n; j++)                  // loop from 0 to n-1
42       for (int k=0; k <= j; k++)               // loop from 0 to j
43         total += first[k];
44     if (second[i] == total) count++;
45   }
46   return count;
47 }
```

Code Fragment 4.12: Some sample algorithms for analysis.