

Exercise Big-O:

<pre>public static void methodA(int n) { int count = 0; for(int i=0; i<20000; i++) { count++; } }</pre>	O(1)
<pre>public static void methodB(int n) { int count = 0; for(int i=0; i<n; i++) { for (int j=0; j<100; j++) { count++; } } }</pre>	O(n)
<pre>public static void methodC(int[] a) { int result = 0; int n = a.length; for (int i=0; i < n; i++) { result = result + a[i]; } }</pre>	O(n)

<pre> public static void methodD(int[][] a) { int result = 0; int n = a.length; for (int i=0; i < n; i++) { for (int j=0; j < i; j++) { result = result + a[i][j]; } } } </pre>	<p>$O(n^2)$</p>
<pre> public static void methodE(int n) { int count = 0; for(int i=0; i<n; i++) { count++; } for (int j=0; j<n; j++) { count++; } } </pre>	<p>$O()$</p>
<pre> public static void methodF(int[] a) { int result = 0; int n = a.length; for (int i=0; i < n; i++) { result += doSomething(a, i); } } private static int doSomething (int a[i], int n) { int result = 0; for (int i=1; i < n; i++) { result += a[i] * n; } return result; } </pre>	<p>$O(n^2)$</p>

