**Running time of algorithms (Big Oh)**

* Computer scientists measure how fast an algorithm runs by counting the number of "basic operations" an algorithm does.
  + Basic operations are things that we consider that a computer can do in "one step," like printing a value, assigning to a variable, simple math operations, etc.
* We then come up with a formula for how many basic operations the algorithm takes, in terms of the algorithm's "input size," which we usually call "n."

// assume array is an array of three ints

// How many operations?

for (int i = 0; i < 3; i++) {

System.out.println(array[i]);

}

// assume array2 is an array of six ints

// How many operations?

for (int i = 0; i < 6; i++) {

System.out.println(array[i]);

}

// How many operations?

for (int i = 0; i < array.length; i++) {

System.out.println(array[i]);

}

// How many operations in the worst case?

for (int i = 0; i < array.length; i++) {

if (array[i] > 10)

System.out.println(array[i]);

}

// algorithm A:

var = var + n;

System.out.println(var);

// algorithm B:

for (int i = 0; i < n; i++) {

var++;  
}

System.out.println(var);

// algorithm C:

int sum = 0;

for (int i = 0; i < array.length; i++) {  
 sum += array[i];  
}

// algorithm D:

int sum = 0;

for (int i = 0; i < array.length; i++) {

if (array[i] > 10) {  
 sum += array[i];

}

System.out.println(sum);  
}