Evan Carey
CS 113
Week 2 Lab 2

Self-Check 1, 3, 4

1. Determine how many times the output statement is executed. Indicate whether the algorithm is $O(n^2)$ or $O(n)$.
   a. $O(n^2)$ – Output statement is executed n^2 times.
   b. $O(n)$ – Output statement is executed $2n$ times.
   c. $O(n^2)$ - Output statement is executed $\frac{n(n+1)}{2}$ times
   d. $O(n^2)$ - Output statement is executed $n - 1$ times

3.

O(f(n))	**f(2000)**	**f(4000)**	**f(8000)**	**f(4000)/f(2000)**	**f(8000)/f(4000)**
a. O(log n)	10.966	11.966	12.966	1.091	1.084
b. O(n)	2000	4000	8000	2	2
c. O(n log n)	21932	47864	103728	2.182	2.167
d. $O(n^2)$	4,000,000	16,000,000	64,000,000	4	4
e. $O(n^3)$	$8.0 * 10^9$	$6.4 * 10^{10}$	$5.12 * 10^{11}$	8	8

4. When possible, values are estimated from figure 2.9. Otherwise, they are calculated (shown in red)

Growth Type	**f(20)**	**f(40)**
Exponential	1,048,576	$1.0995 * 10^{12}$
Cubic	4,000	64,000
Quadratic	2,000	8,000
Log-Linear	2,500	5,000
Linear	2,000	4,000
Logarithmic	1,800	2,000

Programming 1

//CompareGrowth.java output

```
n = 0    y1 = 10    y2 = 2
n = 10    y1 = 1010    y2 = 502
n = 20    y1 = 2010    y2 = 2002
n = 30    y1 = 3010    y2 = 4502
n = 40    y1 = 4010    y2 = 8002
n = 50    y1 = 5010    y2 = 12502
n = 60    y1 = 6010    y2 = 18002
n = 70    y1 = 7010    y2 = 24502
n = 80    y1 = 8010    y2 = 32002
n = 90    y1 = 9010    y2 = 40502
n = 100    y1 = 10010    y2 = 50002
```

This result isn't surprising. Because y2 contains n^2 (while y1 only contains n), as n increases, the constants in y1 and y2 have less impact on their values.