Zack Waller CS130 Tu Thu

1)

f(n) = O(g(n))	A
f(n) = Ω(g(n))	B
f(n) = Θ(g(n))	C
f(n) = o(g(n))	A
f(n) = ɯ(g(n))	C

2)

a) linear - 1.2 x 10^6
b) O(N logN) - 1.8 x 10^7
c) quadratic - 1.2 x 10^8
d) cubic - 1.2 x 10^9

3) 150(10000)*log2(10000) =1.99*10^7
150(1000)*log2(1000) = 1.49 *10^6
150(100)*log2(100) = 99657

10000^2 = 100 000 000
1000^2 = 1 000 000
100^2 = 10 000

a)150(N)Log2(N)
b) N^2
c) N^2

4) Was best times of the was the it

5) The code puts all of the data from the queue and pushes it onto the stack. Once all the data is pushed onto the stack, it is then popped out of the stack and enqueued into the queue.

The final data is the data from the original queue in reverse order and the data from the stack appended to the end.

Queue	Stack
1	3
2	2
3	1

Final Queue
3
2
1
3
2
1

6) Algorithm 1: O(n^2)
Algorithm 2: O(n)

7) Worst Case Running time : O(logN)
Worst Case Space Complexity: O(logN)

9) Worst Case Running Time: O(N)
//Note: Needs to add the running time for List.get() but I don't have the source to review.

10) Runtimes for stack methods.
Push : O(N)
Pop: O(N^2)
Peek: O(N)
isEmpty: O(1)
Size: O(1)