Answer Key for Practice Questions on Complexity (Week 3)

Tutorial Questions

1. O(n)

2.

	Number of steps	Big O
a	$9 + 0.02N^2 + 0.1N$	$O(N^2)$
b	$N^2 + 2N^{-3}$	$O(N^2)$
c	$N! + 100N^{20}$	$O(N!)$
d	$2^N + N!$	$O(N!)$
e	$5N(\log_2 N) + N \times \mathrm{sqrt}(N)$	$O(N^{1.5})$
f	$N^2(\log_2 N) + N(\log_2 N)^2$	$O(N^2 \log N)$
g	$10N^2\log(N) + 5N^3 + N^{\log(N)}$	$O(N^{\log N})$
h	$10^5 + 10^4(\log(N))^2 + 10^3\log(N^2)$	$O((\log N)^2)$

See https://youtu.be/GEQPI5FWjfc

3. The complexities are:
   a. O(n + m) or O(max(n, m))
   b. $O(n^2)$
   c. $O(n^2)$

   See 3(c): https://youtu.be/ezIA4GqDzx8

4. The complexities are:
   a. For f(n), it's O(n)
   b. For g(n), it's $O(n^2)$
   c. For h(n), it's $O(n^2)$

   See https://youtu.be/Pc-kO4IOMHc

5. The complexities:
   a. $O(n^2)$
   b. $O(n^3)$

   See https://youtu.be/alI1ZpUNhic

6. The complexities:
   a. $O(n^3)$
   b. $O(n^4)$
   c. $O(n^3 \log n)$

   See (a) https://youtu.be/LCpce0_-0AA
   (b) https://youtu.be/hvKe-AFm2Vs
   (c) https://youtu.be/TKjHmpCe9H0

Extra Practice Questions

7. O(1)

8.

	Number of steps	Big O
a	$2N^2 + 2N^3 + 3N^4$	$O(N^4)$
b	$N^2 \times 2N^{-3}$	$O(N^{-1})$
c	$N! \times 100N^{20}$	$O(N! \times N^{20})$
d	$5 \times (2N)!$	$O((2N)!)$
e	$N(\log_2 N) + N(\log_3 N) + N(\log_4 N)$	O(N log N)
f	$N(\log_2 (2N))$	O(N log N)
g	$1000 + 5\log(N) + 2N + N^2 + 2^{2N}$	$O(2^{2N})$ or $O(4^N)$ because $(x^a)^b = x^{(ab)}$
h	$\log(N^5) + \log_5(N) + 5N$	O(N)

9. The complexities are:
   a. O(n)
   b. $O(n^2)$
   c. O(nk)

10. The complexities are:
   a. $O(n^2)$
   b. O(n log n)
   c. $O(n^2)$

11. Answers:
   a. $O(2^n)$
   b. O(1)

12. Answers:
   a. These lines set **ith_min** to the next smallest value after **min**.
   So, if a = [1, 2, 3, 4, 5] (position of the elements in a does not matter), max = 5 and min = 1. **ith_min** will be set to 2.
   If min = 2, **ith_min** will be set to 3.
   If min = 3, **ith_min** will be set to 4.
   b. i = 4, ith_min = 6
   c. $O(n^2)$
   d. yes. It is possible to come up with an algorithm with time complexity of O(n log n). sort list first (using merge sort) in ascending order, then grab the kth element in the list using index (k-1).

~End