Chapter 3

1a A finite sequence of precise instructions which are used for solving a problem.

1b By their operational complexity, if it is optimal, if it is greedy.

1c A computer program is based off of an algorithm. The algorithm itself can be ported to different computer programs.

2a Keep track of the largest value while running through the list. Return the largest value at the end.

2b procedure \((a_1, \ldots, a_n)\)

\[
m := a
\]

\[
\text{for } j := 1 \text{ to } n
\]

\[
\text{if } a_j > m
\]

\[
m := a_j
\]

\[
\text{return } m
\]

3a \(f(n) \text{ is } O(g(n)) \iff \exists C, k : n > k \Rightarrow |f(n)| < C|g(n)|\)

3b \(n^2 + 18n + 107 < n^3 + 18n^3 + 107n^3 = 125n^3 \text{ when } n > 1 \text{ So } |f(n)| < 125|n^3|\)

3c \(C|n^2 + 18n + 107| < C|n^3| \text{ when } n > 125 \text{ so } n^3 \text{ is not } O(f(n))\)

4 \((\log(n))^3, \sqrt{n}, 100n + 101, \frac{n^3}{1000000}, 2^n n^2, 3^n, n!\)

5a

7a Linear Search: Start at the beginning of list, continue until object is found.

Binary Search: List is already ordered. Continue choosing the appropriate half remaining until object is found.

7b Linear search is \(O(n)\), and binary search is \(O(\log(n))\)
Binary search is usually faster, but linear search can be faster if the object to find is at the top of the list.

Start at the beginning of the list, continue until a pair of successive objects are in the wrong order, then switch them and start over. Repeat until you are able to reach the end of the list.

(5, 2, 4, 1, 3) (2, 5, 4, 1, 3) (2, 4, 5, 1, 3) (2, 4, 1, 5, 3) (2, 1, 4, 5, 3) (1, 2, 4, 5, 3)

(1, 2, 3, 4, 5)

Expected computational complexity is $O(n^2)$.

2 Chapter 4

1 210 div 17 = 12, 210 mod 17 = 6

2a $a \equiv b \mod 7 \iff a \mod 7 = b \mod 7$

2b $(-11, 17), (-8, -1), (-7, 0)$

2c $10a + 13 \equiv 10a + 20 \equiv -4a + 20 \equiv -4b + 20 \mod 7$

3 Replacing something with something else in the same congruence class changes nothing when in a modulo space. $a + c \equiv b + c \equiv b + d \mod m$

4 Mod value by 16 and save the value as a hexadecimal digit. Then divide the value by 16 and repeat.

5 $(154533)_8, (D95B)_2$

6 $(111010000110)_2, (1011000011101011)_2$

7 Every positive integer has a unique prime factorization.

9a $c = \gcd(a, b) \iff (c | a \land c | b) \land \forall d : (d | a \land d | b)c \geq d$

9b Prime factorization: Good when it is already factored, or easy to factor.

Euclidean algorithm: Always works, good for large values

Bezout Theorem: For constructing situations where the GCD is useful.

9c 1 (use Euclidean algorithm, or show they are relatively prime)

9d $2^33^55^57^3$