Emily Crilley

Analysis of Algorithms - Homework 2

1. T(n) = T(n-1) + T(n-2)
   T(n) = T(n-1) + T(n-2) + T(n-3)
   T(n) = T(n-1) + T(n-2) + T(n-3) + … + T(n-k)
   T(n) = T(n-k)
   $\sum_{i=0}^{n-1} i = n(n-k)$
2. L(a, b) = (f(n; a, b), f(n + 1; a, b))
   L(a,b) = (b, a+b)
   (b, a+b) = (f(n-1); b, a+b), f((n+1)-1; a,b)
   (b, a+b) = (f(1;b,a+b), f(n; a,b))
   (b, a+b) = (b , f(n-1; b, a+b))
   (b, a+b) = (b, a+b)

3. See hw2.py, Time Complexity: O($nlog(n)$)
4.
   a. Pseudo-polynomial time is an algorithm where the worst case complexity is dependent on the numeric value of inputs.
   b. Fib would be a pseudo-polynomial time algorithm because it is dependent on the value of the input.
   c. FibIt would be a pseudo-polynomial time algorithm because it is dependent on the value of the input.
   d. FibPow would not be a pseudo-polynomial time algorithm because it is dependent on the number of inputs, not the numeric value, as seen with the other two functions.
5.
   a. T(n+1) = T(n) + 5
      T(n) = T(n-1) + 5
      T(n) = T(n-2) + 5 + 5
      T(n) = T(n-3) + 5 + 5 + 5
      T(n) = T(n-k) + 5k
      Let k = n
      T(n) = T(n-n) + 5n
      T(n) = 5n

   b. T(n+1) = n + T(n)
      T(n) = (n-1) + T(n-1)
      T(n) = (n-1) + (n-2) + T(n-1)

T(n) = (n-1) + (n-2) + (n-3) + T(n-2)
T(n) = (n-1) + (n-2) + (n-3) + … +(n-(k+1) + T(n-k)
T(n) = (n-1) + (n-2) + (n-3) + … +(n-(n+1) + T(n-n)
T(n) = (n-1) + (n-2) + (n-3) + … +(n-1) + T(0)

$$\sum_{i=0}^{n} i = n(n-1)/2$$

6.

a. T(n + 1) = 2T(n)
T(n) = 2T(n-1)
T(n) = 2(2T(n-2))
T(n) = 2(2(2T(n-3)))
T(n) = 2^k(T(n-k))
T(n) = 2^n(T(n - n))

$$\sum_{i=0}^{n} 2^n(T(n)) = 2^n(n+1)T(n)$$

b. T(n+1) = 2^{n+1} + T(n)
T(x) = 2^x + T(x-1)
T(n) = 2^n + T(n-1)
T(n) = 2^n + ((n-1) + T(n-2))
T(n) = 2^n + ((n-1) + (n-2) + T(n-3))
T(n) = 2^n + ((n-1) + (n-2) + … + (n-(n-1)) + T(0)
T(n) = 2^n + (n-1) + (n-2) + … + 1 + 0

$$\sum_{i=0}^{n} 2^n = 2^n(n+1)$$

7.

a. T(n) = n + T(n/2)
T(x) = x + T(x/2), replace x with n, n-1, n-2
T(n) = n + T(n/2)
T(n) = n + ((n-1) + T((n-1)/2))
T(n) = n + (n-1) + T((n-1)/2) + ((n-2) + T((n-2)/2)
T(n) = n + (n-1) + T((n-1)/2) + ((n-3) + T((n-3)/2) + … + ((n-k) + T((k-n)/2)
T(n) = n + (n-1) + T((n-1)/2) + ((n-3) + T((n-3)/2) + … + ((n-n) + T((n-n)/2)
T(n) = n + (n-1) + T((n-1)/2) + ((n-3) + T((n-3)/2) + … + (0) + T((0)/2)
T(n) = n + (n-1) + T((n-1)/2) + ((n-3) + T((n-3)/2) + … + (0) + T(0)
T(n)= n + (n-1) + T((n-1)/2) + ((n-3) + T((n-3)/2) + … + (0) + 0
T(n) = 2n - 1 or T(2^m) = 2(2^m) - 1

b. T(n) = 1 + T(n/3)
T(n) = 1 + (1 + T((n-1)/3))
T(n) = 1 + 1 + (1 + T((n-2)/3)

$T(n) = 1 + 1 + 1 + (1 + T((n-3)/3))$

$T(n) = k + 1 + (T((n-k)/3))$

$T(n) = n + 1 + (T((n-n)/3))$

$T(n) = n + 1 + (T(0/3))$

$T(n) = n + 1$ or $T(3^m) = 3^m + 1$