Amber Harding
HW2

1. **Theorem 3** *For any* $n \in \mathbb{N}$, $F_n = \frac{1}{\sqrt{5}}(\varphi^n - \hat{\varphi}^n)$, *where* $\varphi = \frac{1+\sqrt{5}}{2}$ *and* $\hat{\varphi} = \frac{1-\sqrt{5}}{2}$.

$$T_F(0) = 0$$
$$T_F(1) = 0$$
$$T_F(n) = 1 + T_F(n-1) + T_F(n-2)$$

a.
b. For n = 0
   i. $((1 + \sqrt{5})^n - (1 - \sqrt{5})^n)/(2^n\sqrt{5})$
   ii. $((1 + \sqrt{5})^0 - (1 - \sqrt{5})^0)/(2^0\sqrt{5})$
   iii. $(1\text{-}1)/(1*\sqrt{5}) = 0$
   iv. $2^{n-1} * C = 0$
   v. N = 0
   vi. 0 = 0
c. For n = 1
   i. $((1 + \sqrt{5})^n - (1 - \sqrt{5})^n)/(2^n\sqrt{5})$
   ii. $((1 + \sqrt{5})^1 - (1 - \sqrt{5})^1)/(2^1\sqrt{5})$
   iii. $((1 + \sqrt{5}) - (1 - \sqrt{5}))/(2\ \sqrt{5})$
   iv. $0 + (2\ \sqrt{5})/(2\ \sqrt{5}) = 1$
   v. $2^{n-1} * C = 0$
   vi. N = 1
   vii. $2^{1-1} = 1$
   viii. 1 = 1
d. For n = 2
   i. $((1 + \sqrt{5})^n - (1 - \sqrt{5})^n)/(2^n\sqrt{5})$
   ii. $((1 + \sqrt{5})^2 - (1 - \sqrt{5})^2)/(2^2\sqrt{5})$
   iii. $((6 + 2\sqrt{5}) - (6 - 2\sqrt{5}))/(4\ \sqrt{5})$
   iv. $(4\ \sqrt{5})/(4\ \sqrt{5}) = 1$
   v. $2^{n-1} * C = 0$
   vi. N = 2

vii. 2^{2-1}= 2

viii. 2 = 2

2. Time Complexity: $O(2^n)$
   a. T(n) = 3 + T(n-1) + T(n-2)
   b. T(n-1)
      i. T(n) = 6 + 2T(n-2) + T(n-3)
   c. T(n-2)
      i. T(n) = 12 + 3T(n-3) + 2T(n-4)
   d. T(n-3)
      i. T(n) = 21 + 5T(n-4) + 3T(n-5)
   e. T(n) = C + T(n-1) + T(n-2)
3. P(n): $L^n(a, b)$= (f(n:a, b), f(n+1: a, b))
   a. For n = 1
      i. L(a,b) = (b, a+b)
      ii. = (b, a+ b)
   b. For n = m
      i. $L^m(a, b)$= (f(m: a, b), f(m + 1: a, b))
      ii. $L^{m+1}(a, b)$= $L(L^m(a, b))$
      iii. = (f(m+1: a, b), f(m+2:a,b)
      iv. p(n) is true for n = m+1
      v. Hence P(n) is true for all n ∈ N
4. (project) Write a function fibPow that takes a natural number *n*, and returns $(L^n(0, 1))_1$.
   a. L = [0,1],[1,1]
   b. Time complexity = O(Log n)
5. *pseudo-polynomial time*.
   a. A pseudo-polynomial algorithm is an algorithm whose worst-case time complexity is **polynomial** in the numeric value of input (not number of inputs).
   b. fib(n) = $O(2^n)$ is not polynomial because it has a Big O time complexity of $O(2^n)$
   c. fibit(n) = O(n) is not polynomial because it has a Big O time complexity of O(n).
   d. fibPow = O(Log n) is not polynomial.
6. T(0) = 0
   a. (a) $T(n + 1) = T(n) + 5$
      i. For k = 1
      ii. T(n+1) = T(n-1) + 5 + 5
      iii. For k = 2
      iv. T(n-1) = T(n-2) + 5 + 5 + 5
      v. For k = 3
      vi. T(n+1) = T(n-3) + 5 + 5 + 5 + 5

      vii. T(n+1) = T(n-K) + (K + 1) + 5
      viii. T(0) = 0
      ix. N - K = 0

x. K = N
xi. For K = N
xii. T(n+1) = T(n-n) + (n+ 1) + 5
xiii. 5n + 5

b. (b) $T(n + 1) = n + T(n)$
   i. For k=1
      1. T(n+1) = T(n-1) + n - 1 + n
   ii. k=2
      1. T(n+1) = T(n-2) + n-2 + n-1 + n
   iii. k=3
      1. T(n+1) = T(n-3) + n-3 + n-1 + n
   iv. T(n+1) = T(n-k) + [n + (n-1) + (n-2) + …. + (n-k)]
   v. T (0) = 0
   vi. N-k = 0
   vii. k=n
   viii. For k=n
   ix. T(n+1) = n(n+1)/2
   x. T(n+1) = $(n^2 + n)/2$

7. $T(0) = 1$
   a. (a) $T(n + 1) = 2T(n)$
      i. For n = k
      ii. T(n+1) = 2^n * T(0)
      iii. T(n+1) = 2^n * 1
      iv. T(N) = O(2^n)
      v. T(n+10) = 2^{n+1} + T(n)
      vi. For n = n-1
      vii. T(n) = + 2^n T(n-1)
      viii. T(n+1) = 2^{n+1} + 2^n + T(n-1).......(1)
      ix. For n = n-2
      x. T(n-1) = 2^{n-1} + T(n-2)
      xi. T(n+1) = 2^{n+1} + 2^n + 2^{n-1} + T(n-2)....(2)
      xii. For n = n-3
      xiii. T(n-2) = 2^{n-2} + T(n-3)
      xiv. T(n+1) = 2^{n+1} + 2^n + 2^{n-1}+ 2^{n-2} + T(n-3).........(3)
      xv. T(n+1) = 2^{n+1} + 2^n + 2^{n-1}+ 2^{n-2} + ...+ 2^{n-k} + T(n-k)
      xvi. 1 + { [2^{n+1} [1 - $(1/2)^n$] / 1- 1/2 }
      xvii. 1 + 2^{n+2} [1 - 1 / 2^n]
      xviii. 1 + 2^{n+2} - 2^2
      xix. 2^{n+2} - 3
      xx. O(2^n)
      xxi. T(n) = O(2^n)

   b. (b) $T(n + 1) = 2^{n+1} + T(n)$
      i. T(n+1) = 2^{n+1} + 2^n + 2^{n-1}+ 2^{n-2} + T(n-3)

ii. T(n+1) = 2^{n+1} + 2^n + 2^{n-1}+ 2^{n-2} + ...+ 2^{n-k} + T(n-k)
iii. T(n+1) = 1 + 2^{n+1} + 2^n + 2^{n-1}+ 2^{n-2} + ...+1
iv. ! + { [2^{n+1} [1 - $(1/2)^n$] / 1- 1/2 }
v. 1 + 2^{n+2} [1 - 1 / 2^n]
vi. 1 + 2^{n+2} - 2^2
vii. 2^{n+2} - 3
viii. O(2^n)
ix. T(n) = O(2^n)

8. T(1) = 1
   a. (a) T(n)=n+T(n/2) (Assume n has the form n=2 .)
      i. T(n) + n
      ii. T(n/2*2) + n/2 + n
      iii. T(n/2^2) + n/2 + n
      iv. T(n/2^3) + n/2 + n
      v. T(n/2^m) + n/2^m-1 + n/2^m-2 + …… + n/2^1 + n^0
      vi. n/2^m = 1
      vii. 2^m = n
      viii. log2(2^m) = log2(n)
      ix. M = $log_2 n$
      x. For M = log2n
      xi. T(n/2^$log_2 n$) + n/2^$log_2 n$ + n/2^1 + n/2^0
      xii. T(n) = 2n-1
      xiii. T(n) = 0(n)
   b. (b) T(n)=1+T(n/3) (Assume n has the form n=3 .)
      i. T(n) = T(n/3) + 1
      ii. T(n/3*3) + 1 + 1
      iii. T(n/3^2) + 1 + 1 + 1
      iv. T(n/3^m) + 1 + 1 + 1 ….
      v. n/3^m = 1
      vi. 3^m = n
      vii. $log_3$3^ m = $log_3 n$
      viii. M = $log_3 n$
      ix. For M = $log_3 n$
         1. T(n) = T(n/3^m) + 1 + 1 + 1 + …
         2. T(n/3$log_3 n$) + 1 + 1 + 1 + ….$log_3 n$
         3. T(1) + $log_3 n$
         4. 1 + $log_3 n$
         5. T(n) = 1 + $log_3 n$
         6. T(n) = 0 $log_3 n$