David Hatcher

COT4400

Homework 2

1. Suppose we have some arbitrary and positive integer n.
   1. For the first loop:
      1. Line 2: x = 1;
      2. Line 3: y = 0;
      3. Line 4: True, go into while loop
      4. Line 5: y = y + x; //y = 1
      5. Line 6: x = 3x; //x = 3
      6. Current value of x, x = 3, y = 1, x = 2y + 1 = 2(1) + 1 = 3
         1. So after the first loop x = 2y + 1 is true
   2. Suppose that for the kth iteration x = 2y + 1
      1. For the (k+1)th iteration
      2. Line 5: yk+1 = y + x; // yk+1 = y + (2y + 1) = 3y + 1
      3. Line 6: xk+1 = 3x // xk+1 = 3(2y + 1) = 6y + 3
      4. With these values we can use algebra to prove that x = 2y + 1 is correct
         1. Xk+1 = 6y + 3
         2. 2yk+1 + 1 = 2(3y + 1) = 6y + 2 + 1 = 6y + 3
         3. Thus xk+1 = 2yk+1 + 1 does hold true
   3. Therefore, after every iteration of this loop, x will equal 2y + 1
2. Base Case, n = 11, F(n) = 89, 1.5^n = 86.498
   1. Assume that F(k) >= c(1.5)^k for some k >= 11
      1. F(k+1) = F(k) + F(k-1)
      2. = c(1.5)^k + c(1.5)^(k-1)
      3. = c(1.5 +1)(1.5)^(k-1)
      4. = c(2.5)(1.5)^(k-1)
      5. >= c(2.25)(1.5)^(k-1)
      6. >= c(1.5)^(k+1)
      7. = Omega(c(1.5)^(k+1)), for all n >= 11
      8. Therefore, there exists a constants greater than 0 c and n0, where c = 1 and n0 = 11, such that F(n) >= c1, for all n >= n0
3. h(n) = O(f(n)g(n)), f(n) = O(j(n)), g(n) = O(k(n))
   1. h(n) = O(f(n))O(g(n)) Reverse envelopment
   2. = O(j(n))O(k(n)) Transitive properties
   3. = O(j(n)k(n)) Envelopment
   4. Therefore, h(n) does equal O(j(n)k(n))