Zachary Neeley

Lab 9

Task 1) f(1) – 0 strings, f(2) – 0 strings, f(3) – 1 string: 000 f(4) – 3 strings: 0000, 1000, 0001

Task 2) The last 4 bits of n will always be 1000, because when generating the string, the new strings will always contain 3 consecutive zeros.

Task 3) 2(n-4)-f(n-4)

Task 4) 2f(n-1)+2(n-4)-f(n-4)

Task 5) Java File.

Task 6) g(n-1)

Task 7) g(n-2)

Task 8) g(n-3)

Task 9) 2(n-3)

Task 10) g(n) = g(n-1)+g(n-2)+g(n-3)+2(n-3)

Task 11) Java File.

Task 12)

Task 15)

If a ∈ S, b ∈ S, s ∈ S, then a · (b s) = (a · b).  
If a string (s) has the amount n of a’s and b’s, which ab · (n+1) of a’s and b’s the last two letters in (ab) · s are ab.

If a string (s) If a string (s) has the amount n of a’s and b’s, which (b a) · (n+1) of a’s and b’s will let the last two letters in (b a) · s will be ba.

If a occurring string (t) has an equal amount of a’s and b’s, (s · t). The string will have equal amounts of a’s and b’s

Task 16) Each tree gets generated by adding 1 to the number of leaves in each tree, and 1 to the number of vertices. Because the tree has 1 more leaf than the amount of internal vertex, which every true binary tree has 1 more leaf than internal nodes or vertex.

I get T1 · T2 from the trees T1 and T2, we are assuming that statement will be true. The leaves of T1 · T2 contain the leaves from the tree of T1, which also using the leaves of T2. With the internal vertices of T1 · T2 will be the internal vertices of T1 and T2 and the roots of both trees.

V(T1 · T2)=V(T1)+V(T2)+1

By the induction hypothesis,

v(T1)=v(T1)+1 and v(T2)=v(T2)+1, therefore,

L(T1 · T2)=V(T1)+V(T2)+2

L(T1 · T2)=V(T1 · T2)+1