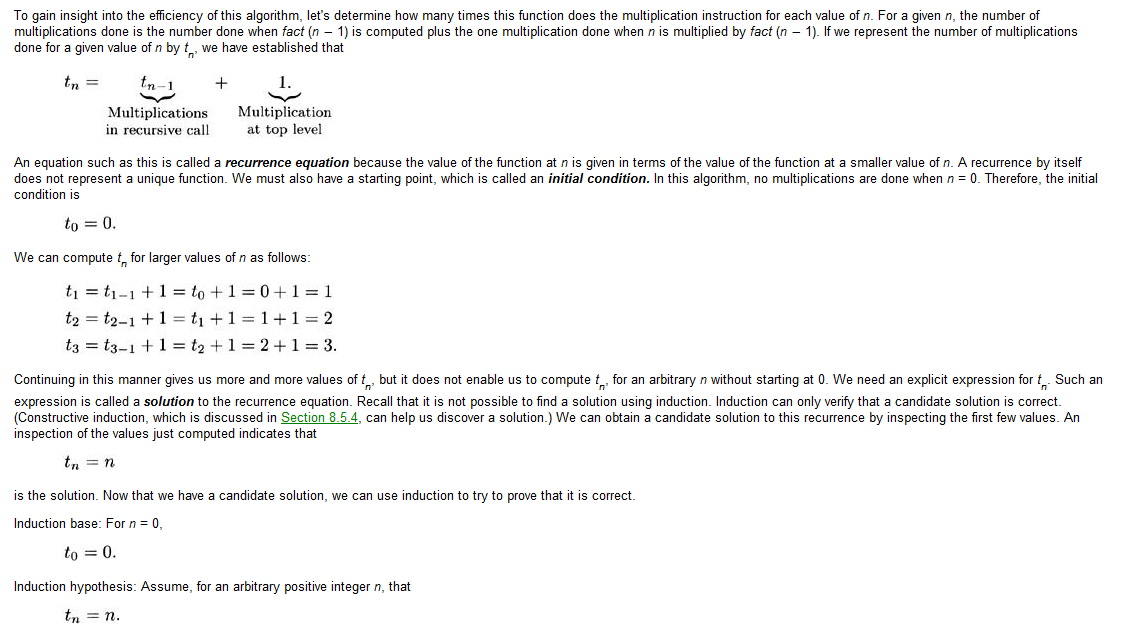
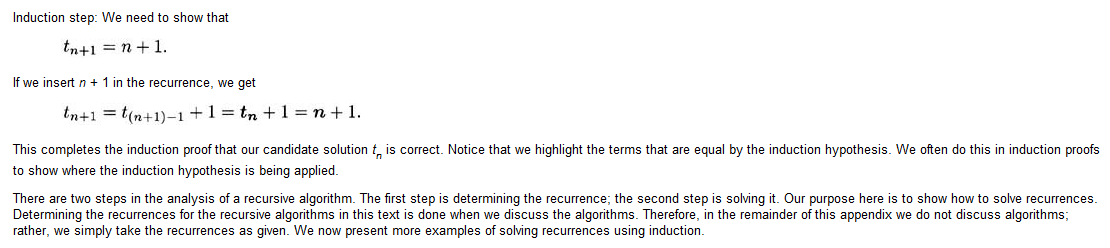
**Factorial (O(n) complexity): (Solving recurrence equation using induction)**





**Fibonacci (O(2n)): (Solving by induction):**

Recurrence Relation of fibonacci = T(n) = T(n-1) + T(n-2) + 1 (the last 1 is the computation of previous two numbers in the series)

for n>= 2 (since n-2 cannot be lower than 0)

Induction Base:

T(2) = T(1) + T(0) + 1 = 1 + 1 + 1 = 3

T(3) = T(2) + T(1) + 1 = 3 + 1 + 1 = 5

T(4) = T(3) + T(2) + 1 = 5 + 3 + 1 = 9

T(5) = T(4) + T(3) + 1 = 9 + 5 + 1 = 15

We can see that for any values of m, the function’s behaviour is similar to 2n

Induction Hypothesis:

T(m) < 2m

Induction Step:

We must show that T(n) < 2n. The value of T(n) is the sum of T(n-1) and T(n-2) plus 1 as we have shown. Therefore,

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

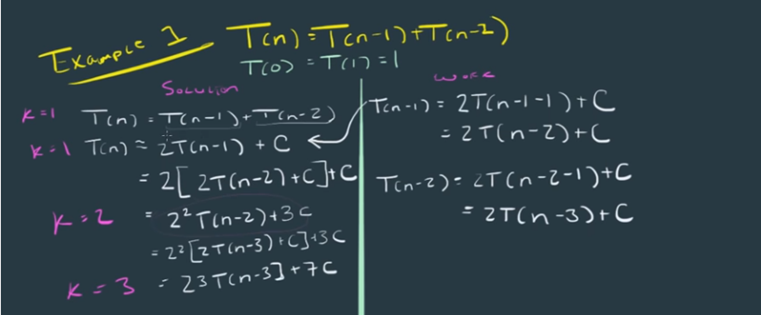
T(n) = 2n-1 + 2n-2 + 1

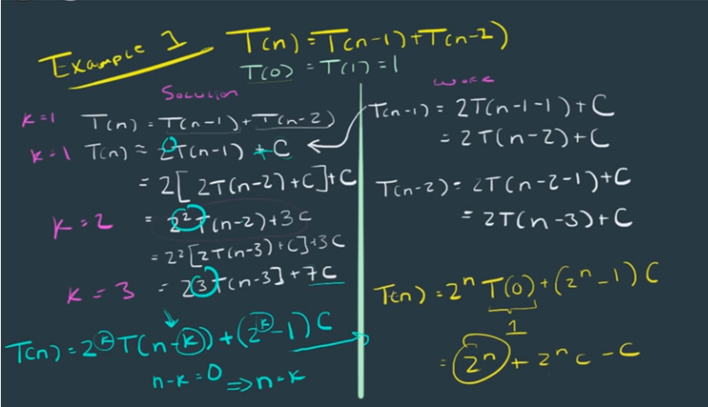
T(n) = (2n-2 ( 2 + 1) )+ 1

T(n) = 2n-2 \* 3 + 1

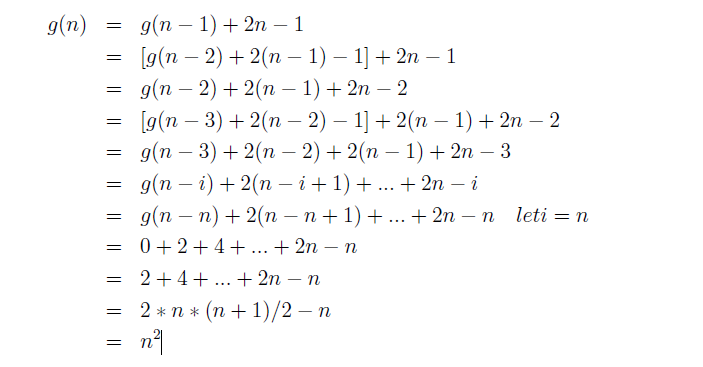
As we can see, the greatest term is 2n-2 which shows that the complexity is O(2n).

**Fibonacci (O(2n)): (Solving by substitution):**





**Substitution example2:**



**MergeSort (Substitution):**

