Noah Hayek (27080409)

Recurrence equations & Big-O

Algorithm	Equations
if n< 4 return 0	n-3 (while the cost is O(1) for one call, this runs n-3 time)
else temp <- A[n-1]*Multiply(A, n-3)	n-1 * (n-3)
Function:	(n-3+n-1)*(n-3) = n^2-7n+12
Big-O	O(n^2)

Tail recursion

Multiply(A. n) does not use tail recursion because the algorithm doesn't end after the last recursive call. In order to be considered tail recursive, the algorithm must finish on the last recursive call. For the given algorithm the last call return 0. From that point onward every returned value must be multiplied with it's preceding one until the stack is cleared. It is possible to design a tail recursive function (see **`TailRecursiveTimingMeasurement.java`**)