Example big-O theorem: Let $f(n) = O(g(n))$ and $t(n) = O(s(n))$, where all functions are positive monotonically growing functions. Prove that $f(n) * t(n) = O(g(n) * s(n))$.

Let c1 and n1 be the constants in the big-O definition for f(n) = O(g(n)) , and c2 and n2 be these constants for t(n) = O(s(n)). In other works, for any n > n1, f(n) ≤ c1*g(n) and for any n > n2, t(n) ≤ c2*s(n). Consider a constant $n_m$= max(n1, n2). For any n> $n_m$, both above inequalities hold. Therefor, since all functions are positive, for any n> $n_m$ f(n)*t(n) ≤ c1*c2*g(n)*s(n)). In other words, we found constants $c_m$ = c1*c2 and $n_m$= max(n1, n2) such that for all n > $n_m$, f(n)*t(n) ≤ $c_m$ *g(n)*s(n)). Then by the definition of big-O, f(n)*t(n) = O(g(n)*s(n)).