constants c and n' such that f(n) <=

For example, let **f(n) = 2n + 5**.

In order to find the Big O of this function, we have to find another function *g(n)* such that *f(n) <= c.g(n)* for all *n >= n'* where *c* and *n'* are positive constants.

If we take *g(n) = 7n* and put this is the above equation, we can see that *(2n+3) <= (7n)* for all *n > 0* and *c = 7*. Hence, we can say that the function *f(n) = O(7n)*. But we write it as *f(n) = O(n)* as we generally analyze an algorithm for a large input *n* and if *n* is large, *n* is approximately equal to *7n*.

The comparison of the function *f(n)* and *g(n)* can also be seen in the graph given below. The function *f(n) <= g(n)* for points where *n*