Chapter VI	Big O								
	Asymptotic notations								
	O(Big O)	Describes an upper bound on time		on the runtime; similar to a less-	then we also say that X<=	In industry, O and theta have been put together and we have to give the tightest description of runtime			
	Omega(n)	Describes the lower bound	for example: printing the values in an array is Omega (n) as well as Omega(logn) as well as Omega(1)						
	Theta(n)	Describes the tight bound on runtime	Theta here means both O and Omega; in this example, it would be Theta(n)						
	Best Case, Worst Case and Expected Case								
	Best Case:	For example, in Quick Sort, if all the elements are equal, then quick sort will, on average, just traverse through the array once - O(N) time	elements greater than pivot - this gives partial sort. then it recursively sorts the left and						
	Worst Case:	The pivot could be repeatedly the biggest element in the array. If pivot is the first element in a reversely sorted array. In this cae, our recursion does not divide the array in half and recurse on other half. Instead, it justs shrinks the subarray by 1 element.	Time taken would O(N^2)						
		both the above best and worst conditions would rarely happen; thus we can expect a runtime of O(nlogn)							
		etween Asymptotic notations , Worst Case and Expected							

There is no particular relationship between the two concepts										
Best Case, Worst Case and Expected Case actually describe the big O or big Theta time for particular scenarios whereas these asymptotic notations describe the upper, lower and tight bounds for the runtime										