The number of function calls and depth of recursion depends on the soring algorithm and the how sorted the array is already. For the cases mentioned in 1b and 1c, number of function calls for sorting a 100 element array will be as follow –

Case – 1b (merge sort algorithm)

Array length = 100

The depth of recursion will be seven. Merge sort divides an array into two parts and keep on running recursion till a part only has one element each. So if we divide 100 by two every time, we get to single element array in seven iterations. Total function calls will be 297.

Case 1b- (quick sort with left pivot and ascending data)

The depth of recursion in this case will be 99. It is because the data is already in ascending order and the left-most pivot strategy only sorts one element at a time. So, quicksort actually becomes a sorting algorithm with O(n2) complexity. Total number of function calls in this case is 298.