1) a) Situations in which the quicksort algorithm is in the best case;
i)When the partition process always picks the middle element as the pivot

b) Situations in which the quicksort algorithm is in the worst case;
i) if the input array is such that, the maximum or minimum element is always chosen as pivot
ii) When the array is already sorted in same order
iii) When the array is already sorted in reverse order
iv) When all the elements are same

2)a) If the subarray has an odd number of elements and the pivot is right in the middle after partitioning, and each partition has (n-1)/2 elements. The latter case occurs when the subarray has an even number n of elements and one partition has at most n/2 elements with the other having n/2-1. In either of these cases, each partition has at most n/2 elements.
We have T(n) = 2T(n/2) + $O(n)$
Using the big theta notation, we get $O(n \log n)$

b) Time complexity of the worst case scenarios of the quicksort algorithm
T(n) = T(0) – T(n-1) + O(n)
Which is equivalent to
T(n) = T(n -1) + O(n)
=$O(n^2)$