Question 5

After looking through the different algorithms I have ranked them like this:

1. Quick Sort - O(nlogn)
2.Merge Sort - O(nlogn)
3.Shell Sort - O(n^2)
4.Insertion Sort -O(n^2)
5.Selection Sort-O(n^2)
6.Bubble Sort- O(n^2)

I ranked these algorithms this way mainly due to their asymptotic complexity. Merge and ,Quick, give us an optimal time complexity of O(nlogn). I ranked merge sort under quick though because merge sort depending on size can operate slower than quick sort, but still operates faster than shell sort which is why I put it ahead of shell sort. In terms of the last 3 sorts, they all have a time complexity of O(n^2), but ranked insertion ahead of selection and bubble, because when tending to small or nearly sorted arrays it operates way faster than selection and bubble sort.