Question 3

a)
For the best case:

k = 1, therefore the outer loop would only run once as on the second iteration we would have i = 1 which would be equal to k and so we would be finished.

outer = 1

Now the inner loop will search from j (i + 1) to n times with the body running n - 1 times.

inner = n - 1 or simply n

Finally all of the swapping we see occuring after the inner loop are all constant factors which we can ignore in asymptotic complexity analysis. This then gives us a total time of:
T(n) = 1 + n, which is just n so we have O(n) complexity for the best case.

b)
For the worst case:

k = n, therefore the outer for loop would run n + 1 times with the body of the loop running n times.

outer = n

Now the inner loop runs the same as in the above question, it will search from j (i + 1) to n times with the body running n - 1 times except that each time it will run 1 time less for each increase of i as j = i + 1 and i is increasing one step per loop so it progressively is getting smaller and smaller each iteration, therefore we get the summation of (n-1) + (n-2) + (n-3) ... 1 =n(n-1) / 2 (the sum of the first n natural numbers) this simplifies down to n^2.

inner = n^2

We can forget about all lower-order terms (the outer loop of complexity O(n)) and constant factors, as we only care about the leading term in asymptotic complexity analysis. This leaves us with T(n) = n^2 + n, we take the leading term and get O(n^2) for our worst case.

Question 4

Since the key is always found exactly in the middle of the array of length n (where n is odd) we can say that:

i = (n + 1) / 2, as the nth term plus 1 then divided by two would give us the middle of the array.

Therefore:

Xi = 1 if i
0 otherwise

The only position that has a probability of 1 is the middle of the array and every other position has a probability of 0.

Therefore:

E[Xi] = 1 * (n + 1) / 2
= (n + 1) / 2

This will always be the average number of cases (n + 1 / 2 comparisons) as the key is always found in the middle of the array. This translates to O(n) for an asymptotic analysis when we remove any constants. This makes sense as regardless of the side of n we will always have to search through half of the array on average before we find the key.