John
Levy

CSE 202
Homework 6

1) First of all we have to assume that all keys are equal. Then whatever test replaced by a strict inequalities will make the loop that starts right before the test run until the other extremity of the subarray which puts the pivot at the other end of the subarray.
Once we notice this, we look at the recursive call will be working on a subarray of lenght $n-1$, which is the worst case we can have with quicksort.
Now let us look at the situation with k distinct keys. At least one occurs at least n/k times. During the recursive calls we will enter at some point with an array of lenght $n/2k$ which gives complexity $O(n^2/(2k)^2) = O(n^2)$ because we have previously assumed that k is fixed.

2) If the array keys can only take two distinct values, then 3-way partioning is sufficient to sort it, so the complexity of quicksort becames linear in n.

3) If we consider the lower bound, that means we are looking at the worst case over $n!$ possibilities of permutations for n elements. In our case however, the worst case is considered over a much smaller input. We also have some additional information regarding the distinct number of keys, so it is compatible that complexity is better than the general worstcase.