Q1: In the worst case, how many guesses would our guessing game take to get the right answer if we had no hints at all?

In the worst case, if we have to guess an integer which is between 1 to 10, then we might have to guess ten times to get the right answer if each guess is not identical to any previous guess. The most times of guesses equals to the range of the integer because each guess is a random event.

Q2: In the worst case, how many guess does it take to get the right number if we get a hint of "higher or lower" when guessing numbers 1-10 and guess intelligently (always picking in the middle of the remaining set of numbers)?

If we give hints of "higher or lower" for each guess, and always pick in the middle of the remaining set of numbers, the largest number of guesses will be log(2)+1 rounded down. Because after each guess, the problem size will be reduced by half, so 2^k=n, then the solution is k=log2(n). log2(10)+1=4.32, rounded it down will be 4. Thus the largest number of guesses to get a correct number between 1 to 10 is 4.

This method is called binary search or logarithmic search. The search process starts at the middle element of the array, and ends if the middle element is exactly the element to be found; if a particular element is greater or smaller than the middle element, the search is done in the half of the array that is greater or smaller than the middle element, and then start the comparison from the middle element as at the beginning. This search algorithm reduces the search range by half for each comparison.