1. Running time: O(n)

   Explanation: This loop runs 2n-1 times. Each time the loop runs, 2 instructions are executed (header + body). The total number of instructions is 2*(2n-1) + 2 = O(n).

2. Running time: $O(n^2)$

   Explanation: This loop runs $n^2 - 1$ times. Each time the loop runes, 2 instructions are executed (header + body). The total number of instructions is $2*(n^2 - 1) + 2 = O(n^2)$.

3. Running time: O(n)

   Explanation: This loop runs 2n-1 times. Each time the loop runs, 2 instructions are executed (header + body). The total number of instructions is 2*(2n-1) + 2 = O(n).

4. Running time: $O(n^2)$

   Explanation: This nested loop incorporates two linear searches that both depends on n (the inner loop depend on i which depends on n). This creates a compound growth. Ignoring all the constants and factors, the result is $O(n^2)$.

5. Running time: O(n)

   Explanation: The outer loop run 99 times. The inner loop run n-1 times. The total number of instruction is 99(n-1) +3. This simplifies to O(n).

6. Running time: O(n*log(n))

   Explanation: The inner loop executes log(n) times. The outer loop executes the log(n) script n times. The result runtime is therefore n*log(n).

7. Running time: O(n)

   Explanation: There are two independent loops which means the total runtime is the sum of the two terms. For Big-O notation, only the fastest growing term is accounted for. Both algorithms are linear, the bound is therefore O(n).

8. Running time: O(1)

   Explanation: There is a constant (499) number of executions of this loop. The Big-O interprets that as a constant input. The answer is, therefore, O(1).

9. Running time: O(n)

   Explanation: The while executes n times. It is simply a while form of a linear for loop. The Big-O bound is therefore O(n).

10. Running time: $O(n^2)$

    Explanation: There are two separate terms. The first term is a nested loop while the second is a linear loop. Since Big-O notation only take accounts for the fastest growing term. Only the quadratic term is considered in this case. The Big-O bound is therefore $O(n^2)$.