Que 4

The time complexity of the provided code is O(n log n).

Explanation:

The outer loop runs from n/2 to n, which means it iterates n/2 times in the worst case.

The inner loop iterates over values of j starting from 2 and doubling in each iteration, up to n. This behaviour is similar to a logarithmic progression.

Combining the two loops, you have a nested loop structure where the outer loop iterates n/2 times, and the inner loop performs a logarithmic number of iterations, which is proportional to log(n). Therefore, the overall time complexity of the code is O(n \* log(n)).

Que 5

The time complexity of the provided code is O(n^2).

Explanation:

The outer loop runs from 0 to n-1, which means it iterates n times.

The inner loop runs from 0 to i-1 for each value of i in the outer loop. In the worst case, when i is at its maximum value of n-1, the inner loop will iterate n-1 times.

Since the inner loop is nested inside the outer loop, you have a quadratic relationship between the number of iterations and the input size n, resulting in a time complexity of O(n^2).