

Exercise 1

a. No, they do not have the same worst case time complexity.

First $O(n*n)$, $O(n+n)$. Therefore the nested for loop case has quadratic time complexity whereas the second snippet has a linear time complexity.

b. The first time through j runs for $a-1$ steps, second time, it is $a-2$ time, third $a-3$ times and so on. Therefore the total number of steps is $(a-1) + (a-2) + (a-3) + \dots + 2 + 1$. Which results in a total complexity of $O(a(a-1)/2)$ which leads to a worst time complexity of $O(a^2)$.

c. In this code, $f(n)$ calls $n(n-1)$ twice and $f(n-1)$ calls $f(n-2)$ twice until we get down to $f(1)$. The resulting structure has a depth of N and thus each level will have twice as many calls as the one above it. The complexity is thus $O(2^N)$.

d. Here the number of elements in the problem gets halved each time, which results in $O(\log N)$ runtime complexity.