Anviti Singh

1002249427

DAA Handson 3

Q1

```
function x = f(n)
x = 1;
for i = 1:n
    for j = 1:n
        x = x+1;
```

Ans (1) • Inner loop runs 'n' times for each iteration of the outer loop.
• Outer loops runs 'n' times.

Total executions $\Rightarrow n \times n = n^2$

Time complexity $\Rightarrow O(n^2)$

Mathematics $T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} 1 = n \times n = n^2$

(3) • Big O → runtime grows quadratically with n $\Rightarrow O(n^2)$
• Big Omega → best case runtime $\Omega(n^2)$, no cases where the function runs in less than n^2 times.
• Big Theta → from the above two the fuction is $\Theta(n^2)$, [both $O(n^2)$ and $\Omega(n^2)$]

modified function

```
x = f(n)
x = 1;
y = 1;
for i = 1:n
    for j = 1:n
        x = x+1;
    y = i+j;
```

(4) The modified function will take slightly more time to run because of the added operation $y = i+j$, within the loop

(5) No, adding the line $y = i + j$ will not affect the result from the runtime analysis in terms of asymptotic notation because the operation is $O(1)$.
Runtime remains $O(n^2)$

Big O : $O(n^2)$
Big Omega : $\Omega(n^2)$
Big theta : $\Theta(n^2)$