UTA ID: 1002240528

Name: Karam Arpan Thakkar

Hands on 3

function $x = f(n)$	cost	time
$x = 1;$	c_1	1
for $i = 1:n$	c_2	n
for $j = 1:n$	c_3	n
$x = x+1;$		1

1. Find the runtime of algorithm Mathematically

Ans So here outer loop runs n times
inner loop runs n time
For each iteration of inner loop $x = x+1$

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

Total no. of operation n^2 Complexity $O(n^2)$

2. In github

3. Find polynomials that are upper and lower bounds on your curve from #2

Ans So as the Curve is quadratic i.e ax^2+bx+c

So they have same upper and lower bound.

As the

Big-O Notation (O): the polynomial is $ax^2 + bx + c$ so time complexity is $O(n^2)$

Big-Omega Notation (Ω): So this represent the lower Bound on time complexity so it grows at least quadraticly respect to n so ~~O(n)~~ $\Omega(n^2)$

Big theta: Time complexity $\Theta(n^2)$
(Average bound) $c_1 g(n) \le f(n) \le c_2 g(n)$
This represents both upper and lower bound
So it grows Qudratically exactly with respect to n

(4.) In github

In modified the function to be:

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

4. ~~with the~~

4. will this ~~affect~~ increate how long it takes the algorithm to run.

Ans So the time complexity will remain same i.e $O(n^2)$

but if we compare with result from # 1 it may take sometime more due to additional operation $y = i + j$ so

It due to computational overhead due to $y = i + j$ at every iteration of inner loop so it will have runtime slightly higher comp

5. will it effect your results from # 1

So it will not affect the results as the time complexity will remain the same $O(n^2)$ so the arithmetically does not affect the complexity