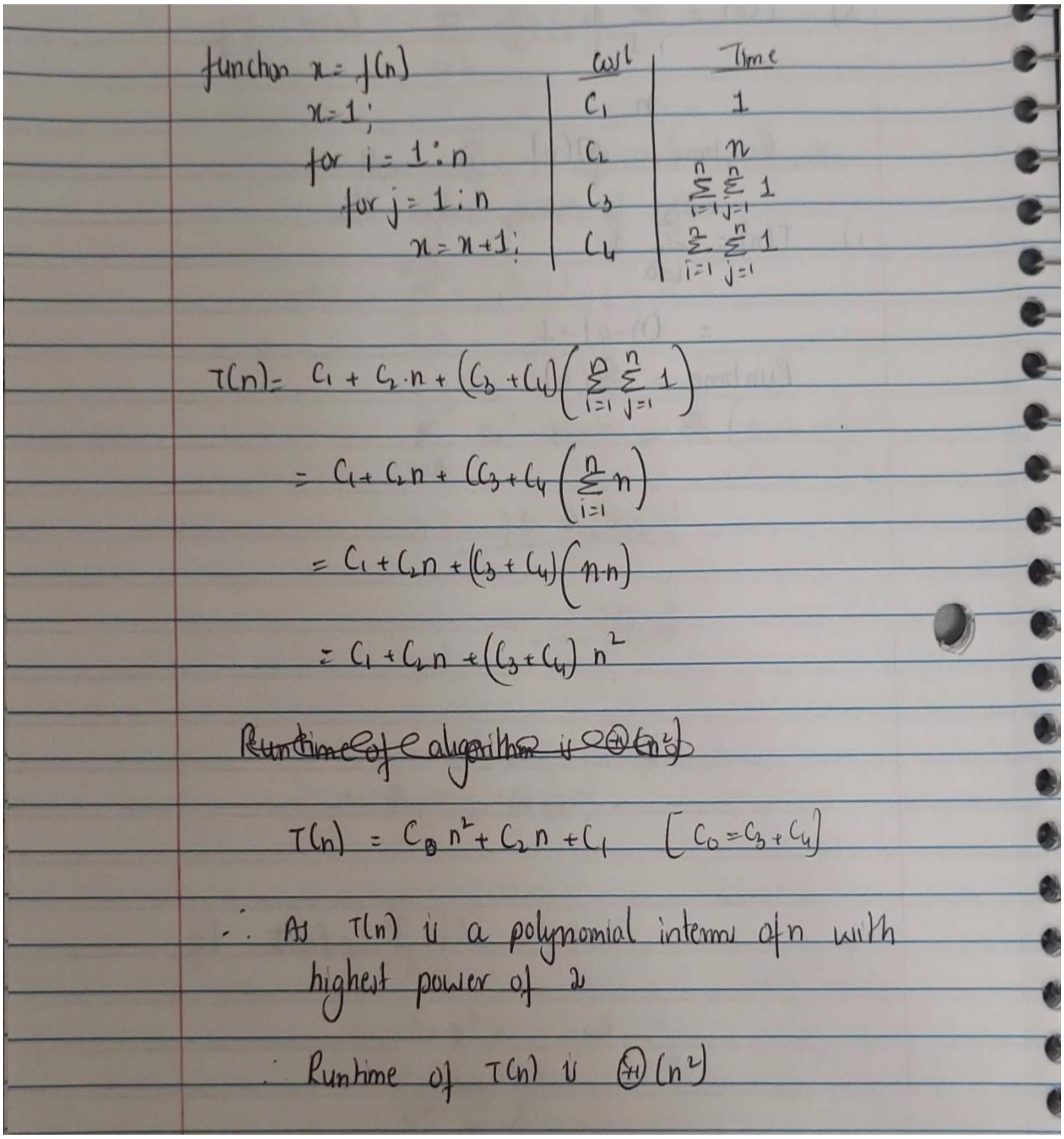
HandsOn-3

1. **Find the runtime of the algorithm mathematically (I should see summations).**



function x = f(n)	cost	Time
x=1;	C_1	1
for i = 1:n	C_2	n
for j = 1:n	C_3	$\sum_{i=1}^{n}\sum_{j=1}^{n} 1$
x = x+1;	C_4	$\sum_{i=1}^{n}\sum_{j=1}^{n} 1$

$$T(n) = C_1 + C_2 \cdot n + (C_3 + C_4)\left(\sum_{i=1}^{n}\sum_{j=1}^{n} 1\right)$$

$$= C_1 + C_2 n + (C_3 + C_4\left(\sum_{i=1}^{n} n\right)$$

$$= C_1 + C_2 n + (C_3 + C_4)(n \cdot n)$$

$$= C_1 + C_2 n + (C_3 + C_4)\, n^2$$

~~Runtime of e algorithm is $\Theta(n^2)$~~

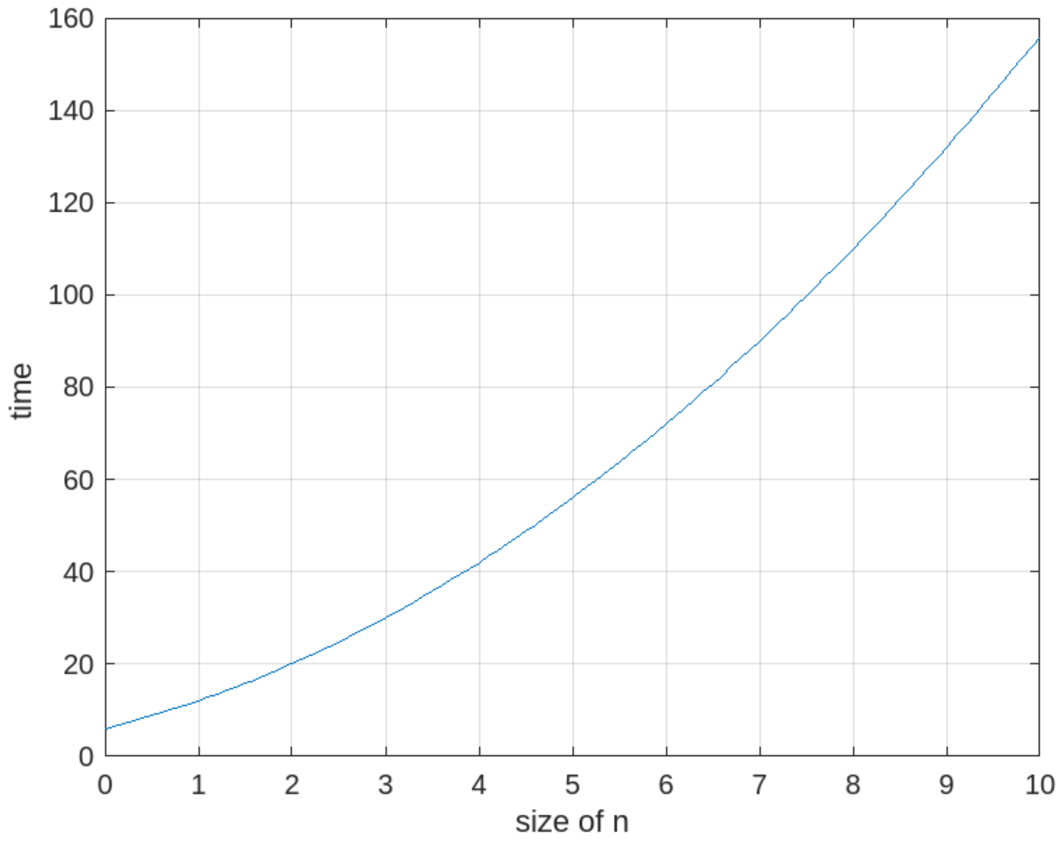
$$T(n) = C_0 n^2 + C_2 n + C_1 \quad [C_0 = C_3 + C_4]$$

$\therefore$ As $T(n)$ is a polynomial in terms of n with highest power of 2

$\therefore$ Runtime of $T(n)$ is $\Theta(n^2)$

2. **Time this function for various n e.g. n = 1,2,3.... You should have small values of n all the way up to large values. Plot "time" vs "n" (time on y-axis and n on x-axis). Also, fit a curve to your data, hint it's a polynomial.**

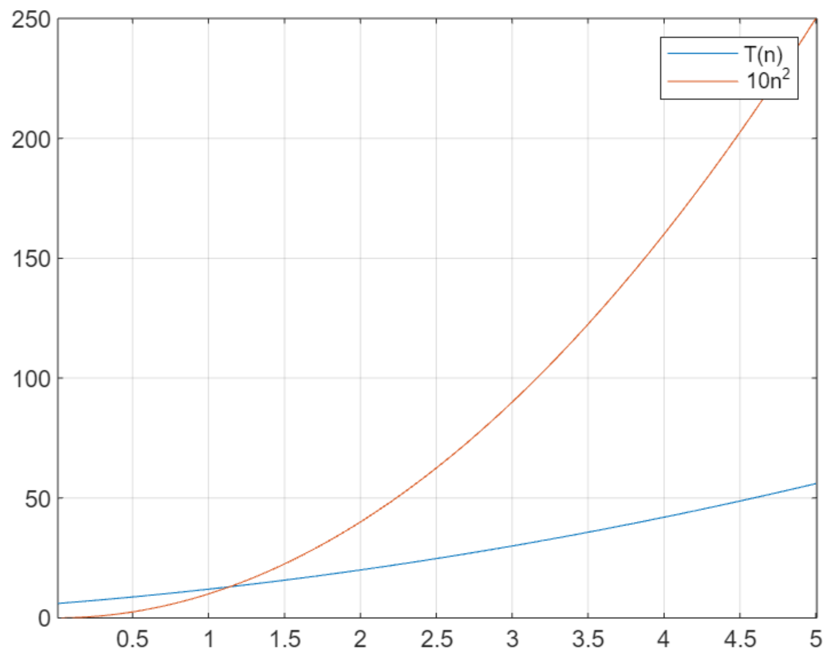
From above we get T(n) = Co n2 + C2 n + C1
Lets assume Co = 1 C1 = 5 C2 = 6
We get $T(n) = n^2 + 5n + 6$
On plotting the graph we get a parabola



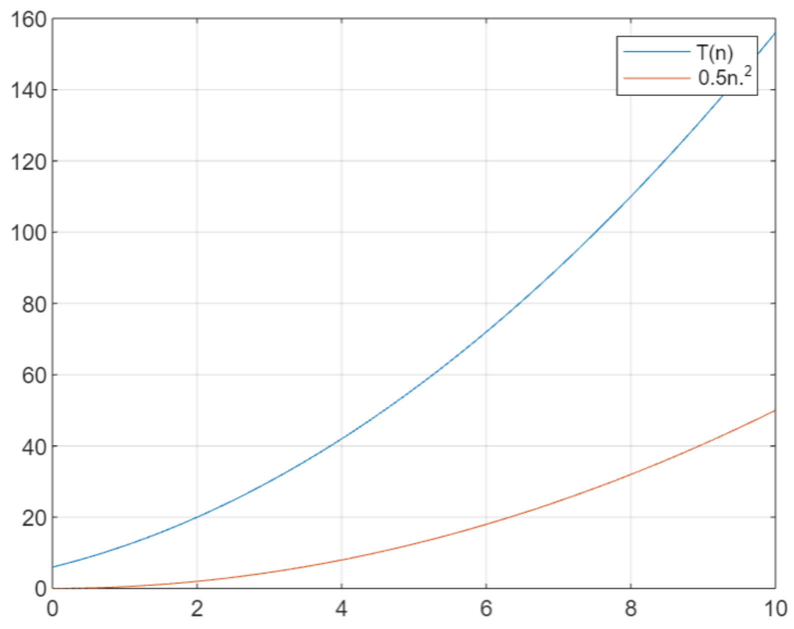
3.Find polynomials that are upper and lower bounds on your curve from #2. From this specify a big-O, a big-Omega, and what big-theta is.

Considering $T(n) = n^2+5n+6$
And upper bound $Cg(n)= 10n^2$
big-O is $O(n^2)$

We get the graph as follows



consider lower bound as $Cg(n) = 0.5n^2$
Big omega is $\Omega(n^2)$
we get graph as follows

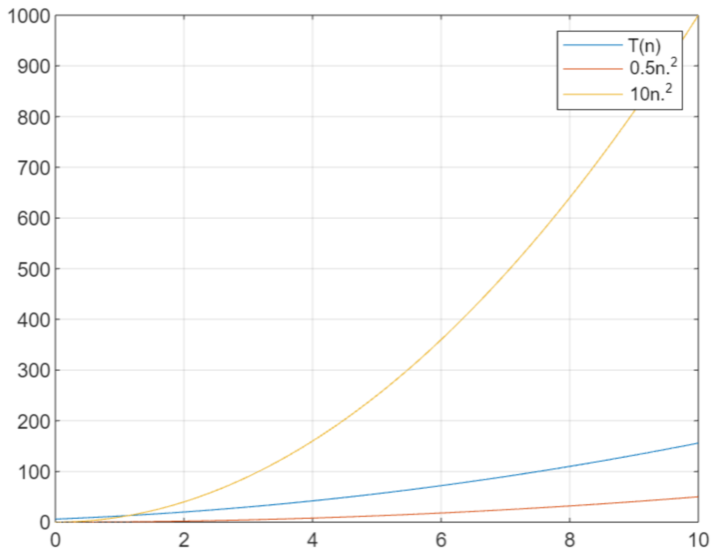


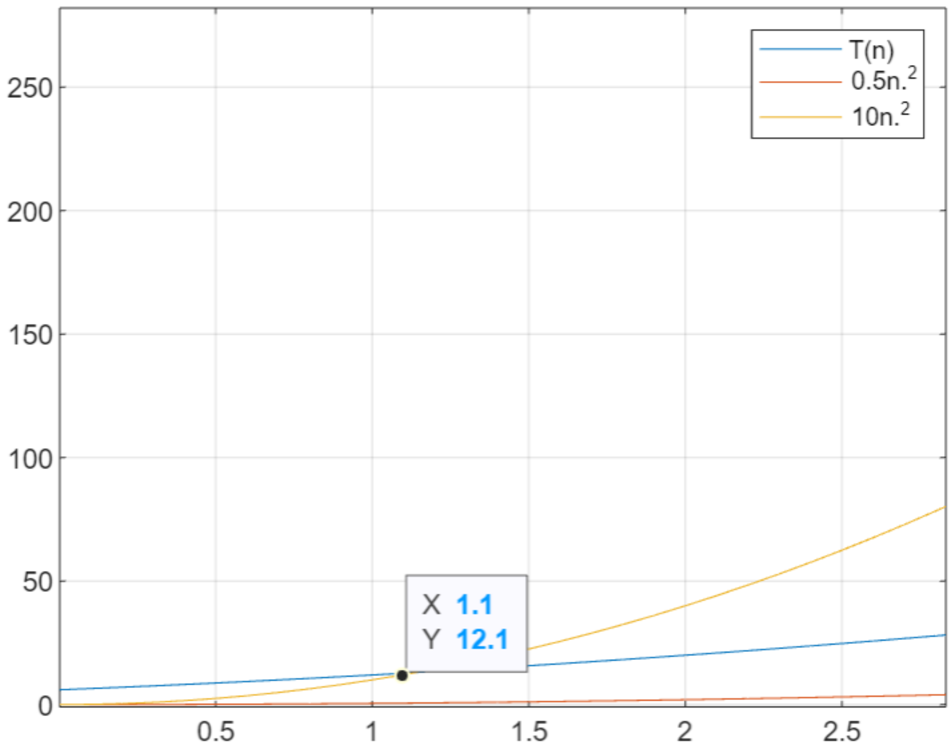
For big theta we have upper bound as $10n^2$ and lower bound as $0.5n^2$ considering the definition of big theta

$\Theta(g(n))$ = { f(n): there exist positive constants c_1, c_2 and n_0
such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0$ }

we can assume big theta as $\Theta(n^2)$

4. Find the approximate (eye ball it) location of "n_0" . Do this by zooming in on your plot and indicating on the plot where n_0 is and why you picked this value. Hint: I should see data that does not follow the trend of the polynomial you determined in #2.





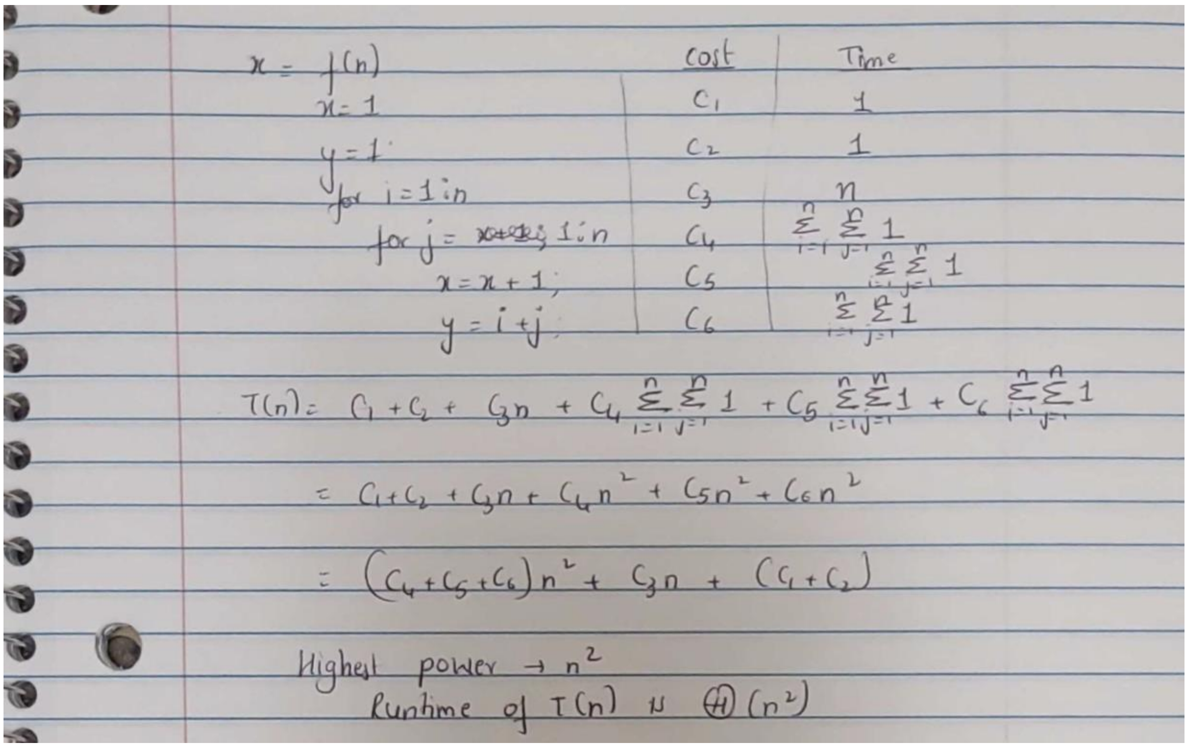
We can consider n0 as 1.1 because for all values of n which are greater than or equal to 1.1. We get $10n^2 <= T(n) <= 0.5n^2$ for all n>=1.1

If I 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. Will this increate how long it takes the algorithm to run

Yes, the modification made to the function will affect the runtime of the algorithm.
As a result, the time complexity of the modified algorithm is same as the original one, the additional operators introduce new factors, but it doesn't change the overall quadratic growth in the runtime with respect to the input size n.



	cost	Time
$x = f(n)$		
$x = 1$	c_1	1
$y = 1$	c_2	1
for $i = 1:n$	c_3	n
for $j =$ ~~x~~ $1:n$	c_4	$\sum_{i=1}^{n}\sum_{j=1}^{n} 1$
$x = x + 1;$	c_5	$\sum_{i=1}^{n}\sum_{j=1}^{n} 1$
$y = i + j;$	c_6	$\sum_{i=1}^{n}\sum_{j=1}^{n} 1$

$$T(n) = C_1 + C_2 + C_3 n + C_4 \sum_{i=1}^{n}\sum_{j=1}^{n} 1 + C_5 \sum_{i=1}^{n}\sum_{j=1}^{n} 1 + C_6 \sum_{i=1}^{n}\sum_{j=1}^{n} 1$$

$$= C_1 + C_2 + C_3 n + C_4 n^2 + C_5 n^2 + C_6 n^2$$

$$= (C_4 + C_5 + C_6) n^2 + C_3 n + (C_1 + C_2)$$

Highest power $\rightarrow n^2$

Runtime of $T(n)$ is $\Theta(n^2)$

5. Will it effect your results from #1

No, As shown in the picture the number of steps of algorithm increased but as we can observe at the end the only change is in the constant values and not the structure of T(n)