Questions – 1

i) Basic operation is the comparison marked as (1)

a) Analyze B(n)

In any case the program executes the first if statement so, the probability to execute the first if statement is 1 so the program is going to execute the basic operation n times so exact complexity is going to be n. but in theta analysis B(n)= Θ(n)

b) Analyze W(n)

In any case the program executes the first if statement so, the probability to execute the first if statement is 1 so the program is going to execute the basic operation n times so exact complexity is going to be n. but in theta analysis W (n)= Θ(n)

c) Analyze A(n)

In any case the program executes the first if statement so, the probability to execute the first if statement is 1 so the program is going to execute the basic operation n times so exact complexity is going to be n. but in theta analysis A (n)= Θ(n)

the input is meaningless because the program is going to execute the first if statement in any case.

ii) Basic operation is the comparison marked as (2)

a) Analyze B(n)

The inputs here is meaningless again since either we are in for j ← i to n − 1 do or for m ← i to n − 1 do so let's all inputs are 1 and we are executing for j ← i to n – 1 so the exact complexity is going to be

n+(n-1)+(n-2)+(n-3)………. 1 = $n * \frac{n+1}{2}$

but in theta analysis $n * \frac{n+1}{2} \epsilon \theta\ (n^2)$

b) Analyze W(n)

the inputs here is meaningless again since either we are in for j ← i to n − 1 do or for m ← i to n − 1 do so let's all inputs are 0 and we are executing for m ←i to n – 1 so the exact complexity is going to be

n+(n-1)+(n-2)+(n-3)………. 1 = $n * \frac{n+1}{2}$ and in theta analysis it is the member of $\epsilon \theta\ (n^2)$

c) Analyze A(n)

We need to consider two input options but the chances of 1 and 0 are the same so the exact complexity is going to be

$$\sum_{i=0}^{n-1}(n-i) * \frac{1}{2} + \sum_{i=0}^{n-1}(n-i) * \frac{1}{2} = \sum_{i=0}^{n-1}(n-i)$$

= n+(n-1) +(n-2) +(n-3) ………. 1 = $n * \frac{n+1}{2}$ and

in theta analysis it is the member of $\epsilon\theta\ (n^2)$

iii) Basic operation is the comparison marked as (3)

a) Analyze B(n)

If the all the inputs are 1 then the basic operation is not going to be executed so θ (0)

b) Analyze W(n)

If the all the inputs are 0 then the basic operation is always executed.

First loop is going to be executed n times.

Second loop is going to be executed $\frac{n*(n+1)}{2}$ times.

First loop is going to be executed $\frac{n*(n+1)*logn}{2}$ times.

The basic operation is going to be executed $\frac{n*(n+1)*logn}{2}$ times.

So the exact complexity is going to be $\frac{n*(n+1)*logn}{2}$

Then in the thata analysis-> (don't care ½ since it is constant) $(n^2+n)logn = n^2logn + nlogn\ and\ \ nlogn\ has\ low\ order\ then\ n^2logn\ so\ \epsilon\ \ \theta\ (n^2logn)$

c) Analyze A(n)

The chance of getting from the list 1 and 0 is equal and the number of executions of the basic operation if all the inputs are 0 is $\frac{n*(n+1)*logn}{2}$ times.

So, we simple multiply this value by probability which is ½ =$\frac{n*(n+1)*logn}{4}$

The basic operation is going to be executed $\frac{n*(n+1)*logn}{2}$ times.

So the exact complexity is going to be $\frac{n*(n+1)*logn}{4}$

and in theta analysis we don't care ¼ since it is constant and

$n*(n+1)*logn = (n^2+n)logn =$

$n^2logn + nlogn\ and\ \ nlogn\ has\ low\ order\ then\ n^2logn\ so\ \epsilon\ \ \theta\ (n^2logn)$

iv) Basic operation is the comparison marked as (4)

a) Analyze B(n)

If all inputs are 1 then the basic operation is going to be less executed since the second y=y+1is going to be executed less in

for m ← i to n − 1 do loop then

The "for m ← i to n – 1" loop will be executed $\frac{n*(n+1)}{2}$ times. Then the complexity will be $\frac{n^2}{2}+\frac{n}{2}$ and theta analysis it is ϵ θ (n^2)

b) Analyze W(n)

If all inputs are 0 then the basic operation is going to be executed more since the first y=y+1is going to be executed more

in for k ← n downto 1 by k ← $\lfloor k/2 \rfloor$ do

If all inputs are 0 then the loop" for k ← n downto 1 by k ← $\lfloor k/2 \rfloor$ do "is going to be executed $\frac{n*(n+1)*logn}{2}$ times. So, exacly the complexity is going to be $\frac{n*(n+1)*logn}{2}$ which is the member of θ $(n^2 logn)$

Consequently, W(n) ϵ θ $(n^2 logn)$

c) Analyze A(n)

we consider two case if the input is 1 or 0 but the chances are equal

$$\frac{1}{2}(Execution\ Times\ as\ inputs\ are\ 1)+\frac{1}{2}(Execution\ Times\ as\ inputs\ are\ 0)$$

So the exact complexity will be

$$\frac{1}{2}*\frac{n*(n+1)*logn}{2}+\frac{1}{2}*\frac{n*(n+1)}{2}=\frac{1}{4}(n^2*logn+nlogn+n^2+n)$$

When we consider theta analysis $don't\ consider\ the\ \frac{1}{4}$ and $n^2*logn\ has\ bigger\ order\ than\ the\ others$

Then the complexity will be A(n) ϵ θ $(n^2 logn)$

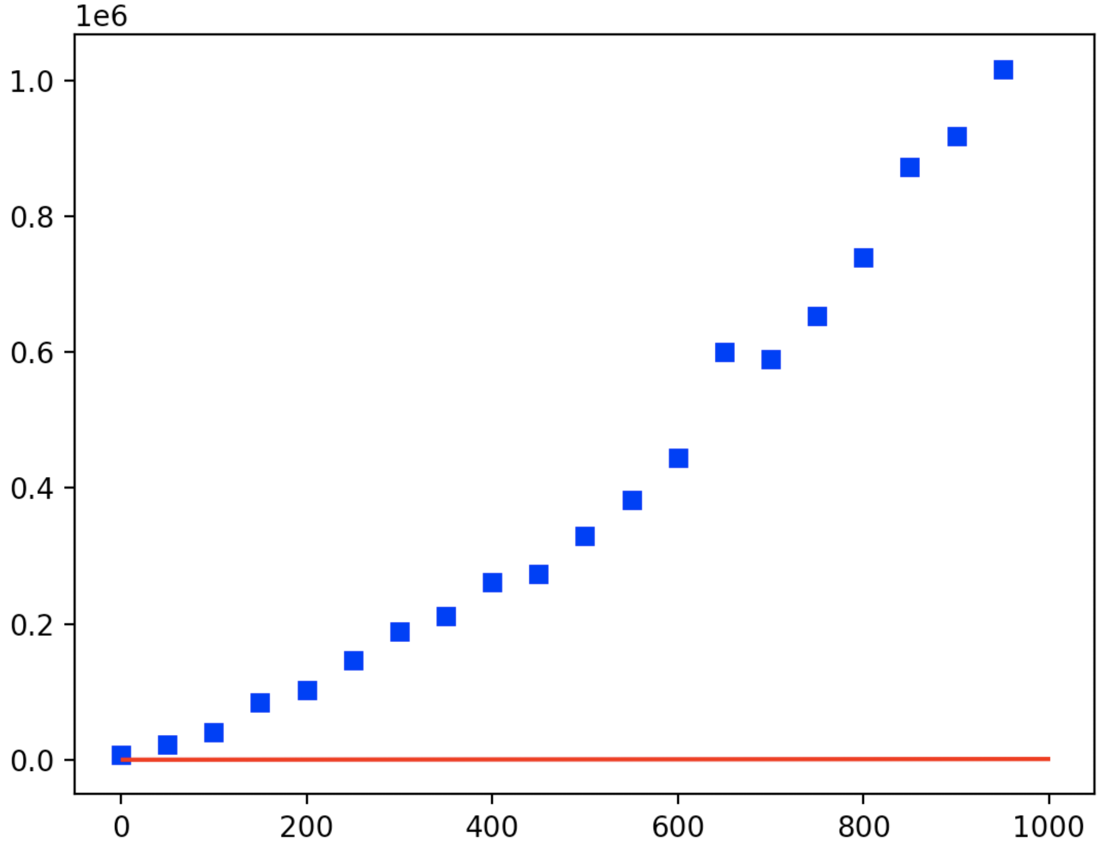
Real Execution:
Clearly, we need to choose basic operation marked as (4) so let's compare the theoretical analysis and our actual real execution one by one.

We have 3 different complexity function as we collect above Θ(n), θ (n^2), θ ($n^2 logn$) and 3 different cases best inputs, worst inputs, and average inputs.

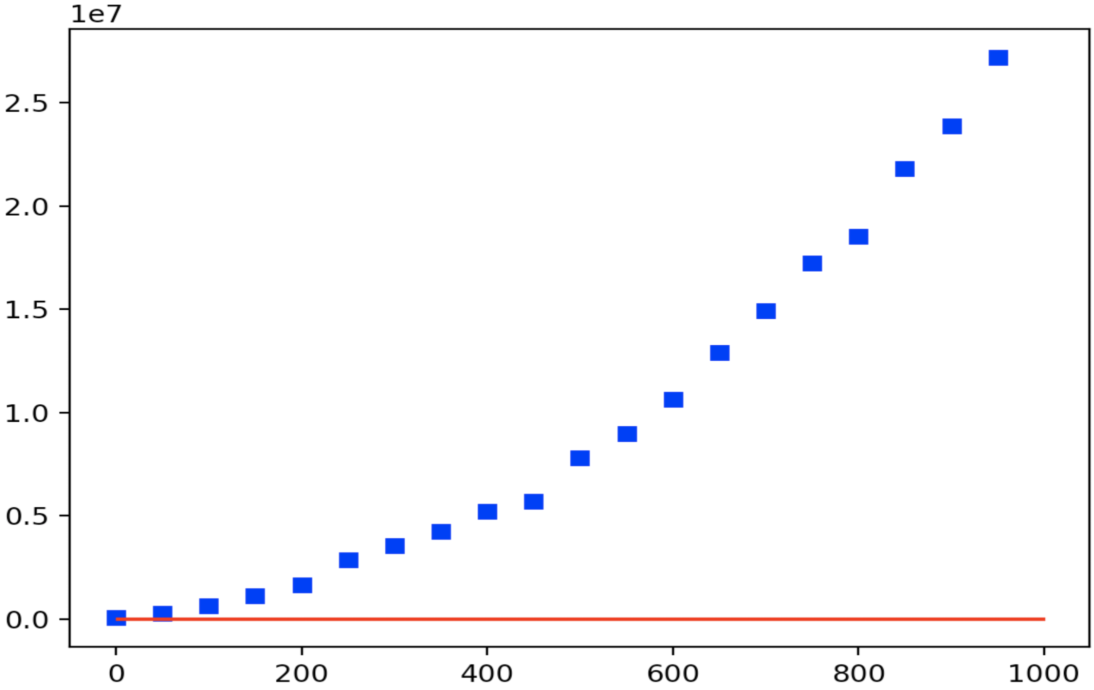
To have best complexity in our program we need to have all inputs 1.
To have best complexity in our program we need to have all inputs 0.

For all these graphs below, in y-axis denotes nanoseconds and x-axis denotes input size. The complexity function is drawn in red color and the samples are drawn in blue.

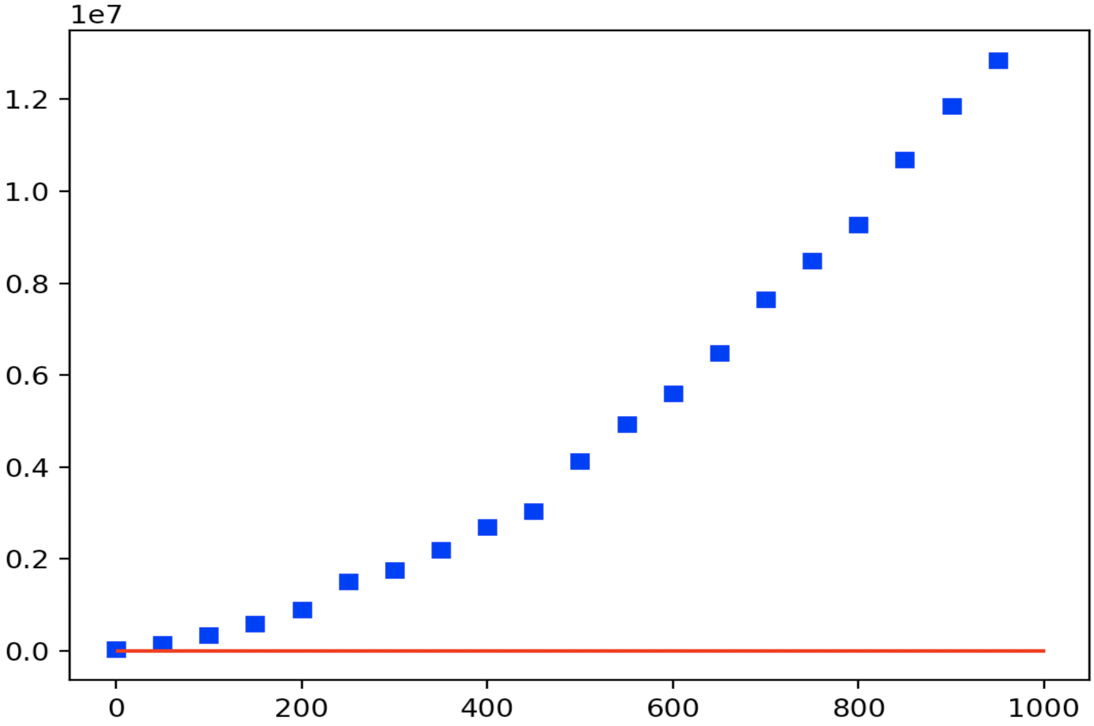
Θ(n) vs Our program with best inputs



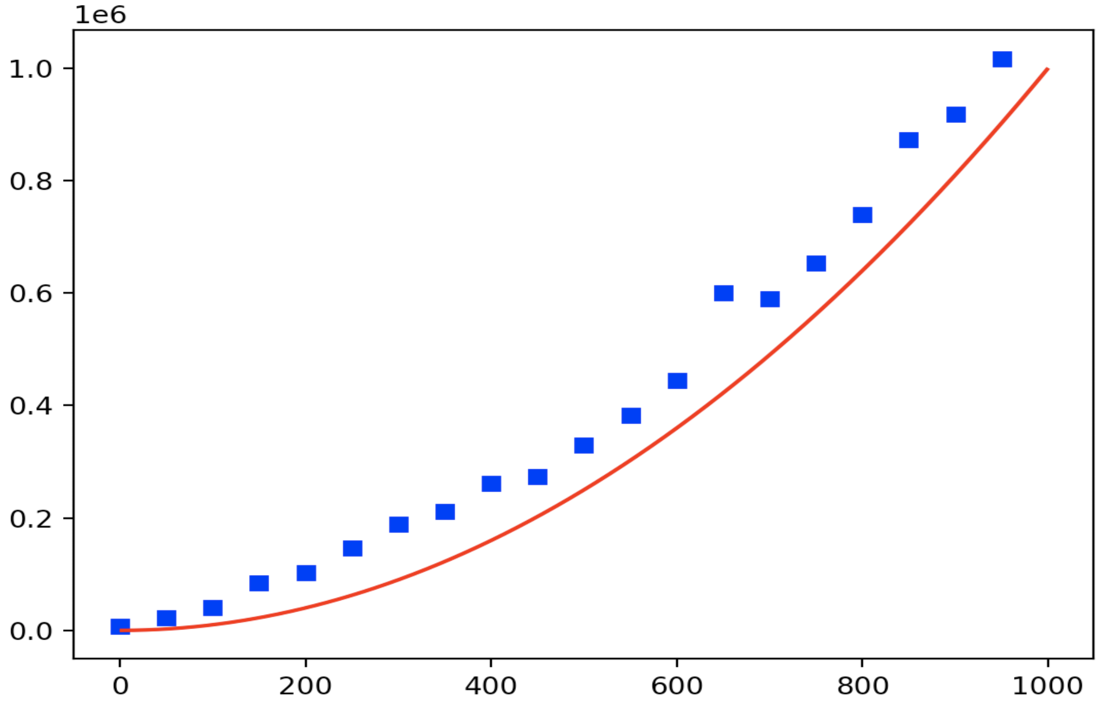
Θ(n) vs Our program with worst inputs



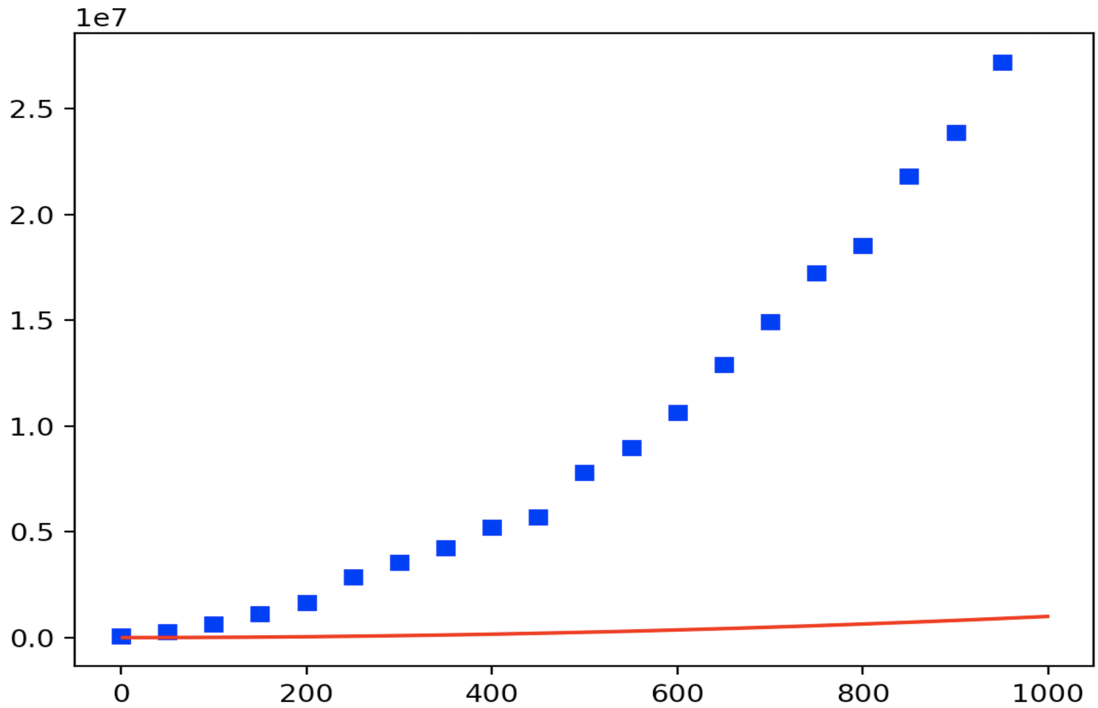
Θ(n) vs Our program with average inputs



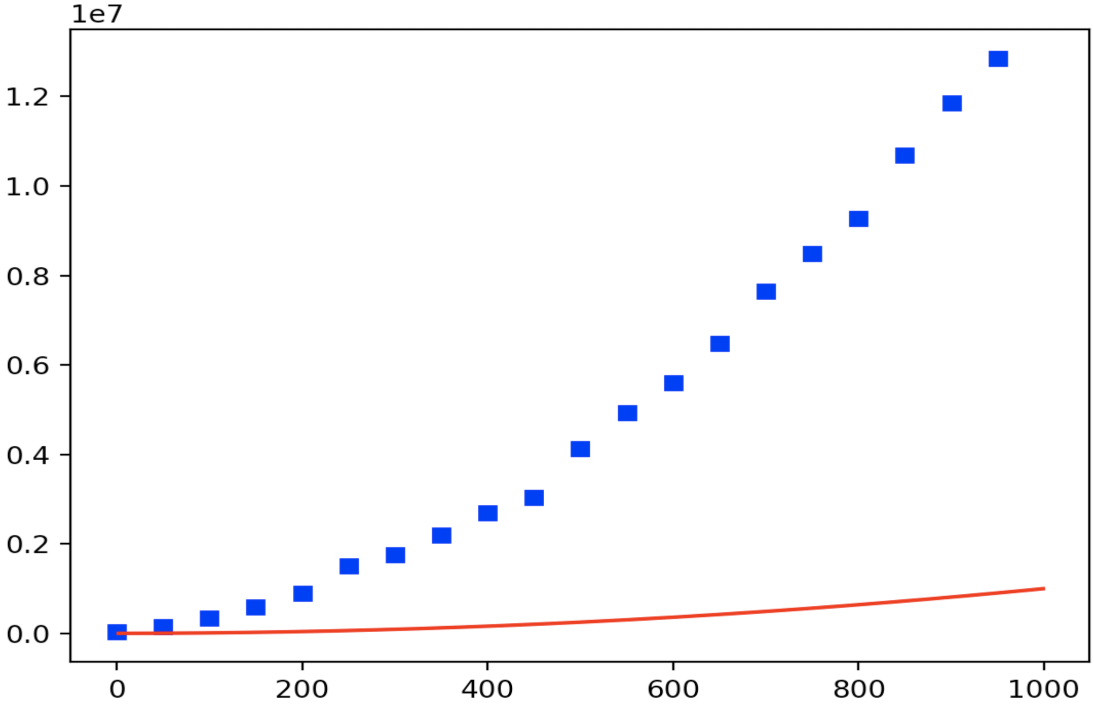
θ (n^2) vs Our program with best inputs



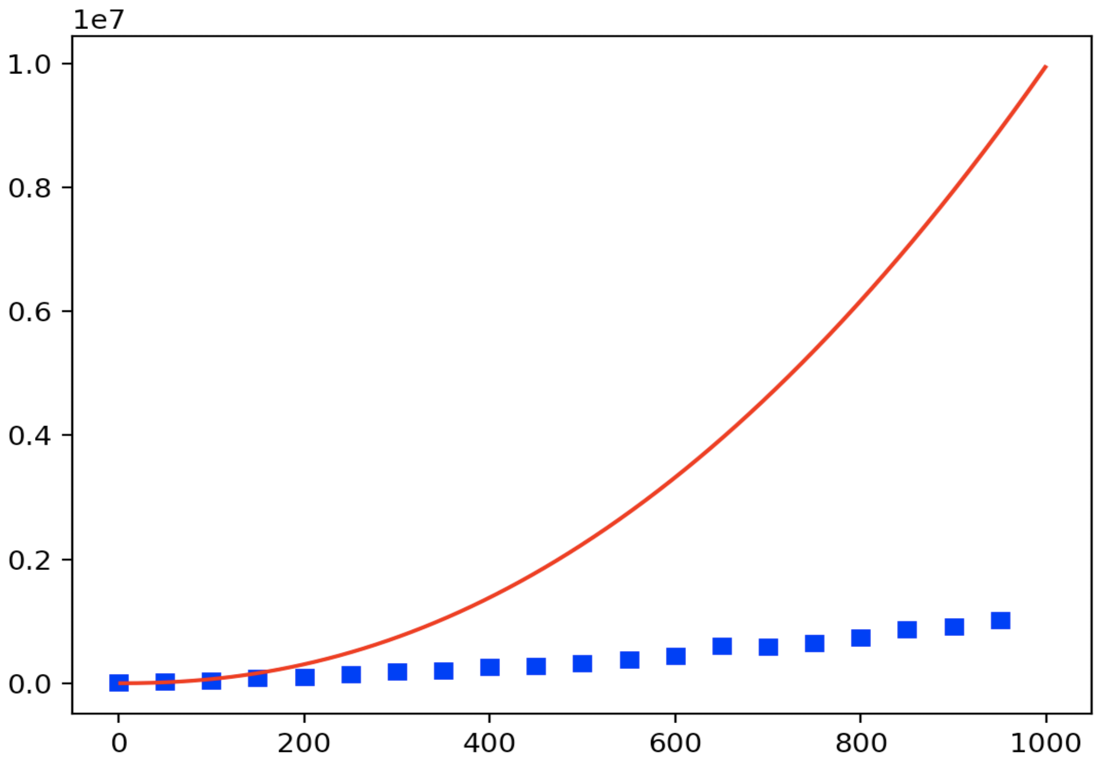
θ (n^2) vs Our program with worst inputs



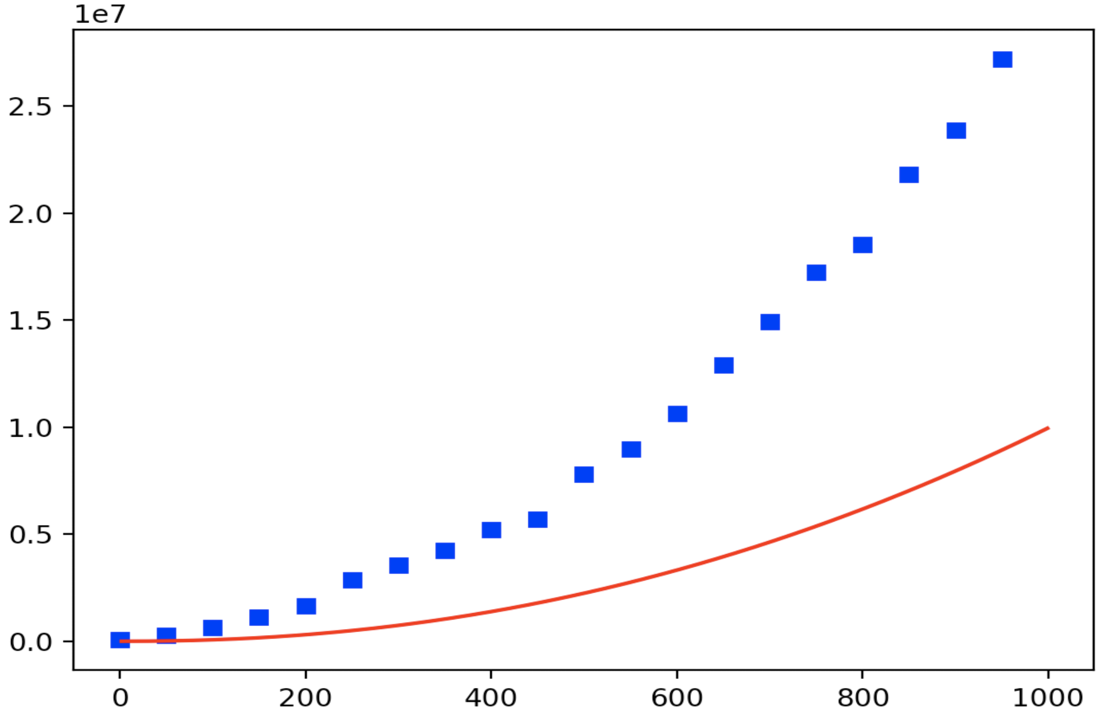
θ (n^2) vs Our program with average inputs



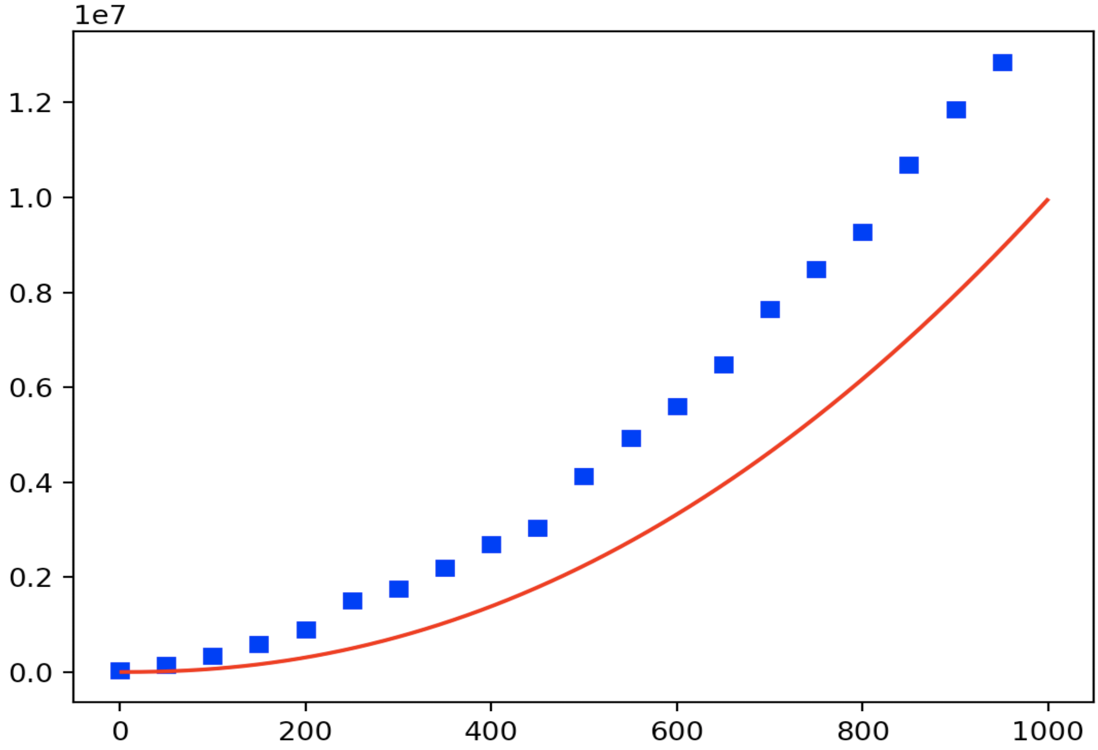
θ ($n^2 logn$) vs Our program with best inputs



θ ($n^2 logn$) vs Our program with worst inputs



θ ($n^2 logn$) vs Our program with average inputs



Then let's compare these complexity functions and how our actual programs react with different inputs.

In case 1:

we have θ (n) for all cases and as we see from the graphs, our actual program with best, worst, and average inputs has no relation with θ (n) so as in case 1 the basic operation cannot be (1)

In case 2:

we have θ (n^2) for all cases and as we see from the graphs our program in best input and θ (n^2) have a relation but θ (n^2) and our program's in worst and average inputs have no relations so as in case 2 the basic operation cannot be (2)

In case 3:

We have no execution in best inputs but in worst and average case the complexity is element of θ ($n^2 logn$). As we can see graphs above, we see that case3's worst and average cases have a relation with our actual program's worst and average cases. However, we still cannot say that case 3 can define our program because in best cases we cannot construct relation in best case.

In case 4:

We have θ (n^2) complexity in best case in case and θ ($n^2 logn$) complexity in worst and average cases and when we look at the related graphs it is clear that we can construct a relation between case 4 and our actual program.

In case 4 best case:

Look at θ (n^2) vs programs best case graph. The complexity functions and plots taken from our programs in best case behave similar.

In case 4 worst case:

Look at θ ($n^2 logn$) vs programs worst case graph. The complexity functions and plots taken from our programs in worst case behave similar.

In case 4 average case:

Look at θ ($n^2 logn$) vs programs average case graph. The complexity functions and plots taken from our programs in average case behave similar.

Consequently, our program has

1. in best case θ (n^2), which case 4 has
2. in best case θ ($n^2 logn$), which case 4 has
3. in best case θ ($n^2 logn$), which case 4 has

All in all, the basic operation is absolutely 4

Questions – 2

a) $f(n)\in\Theta(g(n)) \Leftrightarrow g(n)\in\Theta(f(n))$

→ first

Let f(n)=5n and g(n)=n, Θ(g(n)) is n

And clearly, we can write 5n ϵ Θ(n) and

Θ(f(n)) is again n so n ϵ Θ(n)

If we $\lim_{n\to\infty}\frac{f(n)}{g(n)}$=5 which is constant and

If we $\lim_{n\to\infty}\frac{f(n)}{g(n)}=\frac{1}{5}$ which is constant

← second

Is the same

Consequently, this property holds

b) $f(n)\in o(g(n))\Leftrightarrow g(n)\in\omega(f(n))$

→ first

Let $f(n) = 5^n + n^2 + 5$ *and* $g(n) = 7^n + 2$ so when we take the limit

$\lim_{n\to\infty}\frac{f(n)}{g(n)}$=0

so, first part is satisfying and

when we consider second part and take limit again

$\lim_{n\to\infty}\frac{g(n)}{f(n)}=\infty$

← second

Is the same

Consequently, this property holds

c) $f(n)\in o(g(n))\Rightarrow f(n)\in O(g(n))$

$$let\ f(n) = 5n^2\ and\ g(n) = 6n^3$$

$$so\ we\ can\ write\ first\ part\ f(n)\in o(g(n))$$

Since $\lim_{n\to\infty}\frac{f(n)}{g(n)}$=0

Then let's see if we can write $f(n)\in O(g(n))$,

$5n^2 \le$c*6n^3

$$so\ we\ can\ easly\ say\ c = \frac{5}{6}\ and\ n0 = 1$$

there exist positive constants c and n0 such that integer n0≥n

d) $f(n) \in \omega(g(n)) \Rightarrow f(n) \in \Omega(g(n))$

$$let\ f(n) = n^{-2}\ and\ g(n) = n^{-3}$$

$$so\ we\ can\ write\ first\ part\ f(n) \in\ \omega\ (g(n))$$

Since $\lim_{n\to\infty} \frac{f(n)}{g(n)}=\infty$

So

Then let's see if we can write $f(n) \in\ \Omega\ (g(n))$,

c*$n^{-3} \le n^{-2}$

$$so\ we\ can\ easly\ say\ c = 1\ n0 = 1$$

there exist positive constants c and n0 such that integer n0≥n

e) $\Theta(f(n)) = \Theta(c{*}f(n))$, $where\ c\ is\ a\ positive\ constant$

Let f(n) =$5n^2$ so $\Theta(f(n))$ consist of elements such that cn^2

To prove this property let's take an element of $\Theta(f(n))$ and call it g(n)

Let g(n)= $\frac{n^2}{2}$ and we can write the equation like this

$c1 * 5n^2 \le \frac{n^2}{2} \le c2 * 5n^2$ and we can say n0=1 and c1=1/10 and c2 =2/10

So, first property holds let's look at second part lets multiply f(n) by 10 and write the equation again.

$$c3 * 5 * 10n^2 \le \frac{n^2}{2} \le c4 * 5 * 10 * n^2$$

we can say n0=1 and c1=1/50 and c2 =2/50

so, the overall property holds

f) $f(n) \in O(g(n)) \Leftrightarrow O(f(n)) \subseteq O(g(n))$

→ first

Let $f(n) = 2^n$ and $g(n) = 4^{n^2}$ so $f(n) \in O(g(n))$ holds since

$O(g(n))$ consists of all the multiplies of 4^{n^2} and all the low order functions

So we can write $c * logn, c * n , c * logn , n^2, c * 2^n, and\ c4^{n^2} \in O(\ 4^{n^2})$

Let's consider second property $O(f(n)) \subseteq O(g(n))$

$O(f(n))$ is a cluster and consists of all multiplies of 2^n and low order functions such as $c * logn, c * n , c * logn , n^2 \ldots and\ c * 2^n$

So Clearly

$c * n , c * logn , n^2 \ldots and\ c * 2^n \subseteq c * logn, c * n , c * logn , n^2, c * 2^n, and\ c4^{n^2}$

so, the overall property holds

← second

as we look at reverse side

if $O(f(n)) \subseteq O(g(n))$ is we let again $g(n) = 4^{n^2}$ but this time Let $f(n) = n\ then\ proprery\ holds$

all the low orders of n and multiplies of n ⊆ all the low orders of n and multiplies of 4^{n^2}.

Let's try to write $f(n) \in O(g(n))$ and

we can write $n \leq c4^{n^2}$ exist positive constants c and n0 such as c =1 n0=1

g) $f(n) \in o(g(n)) \Leftrightarrow O(f(n)) \subset O(g(n))$

→ first

$let\ f(n) = logn\ and\ g(n) = n\ so\ we\ can\ write\ f(n) \in o(g(n))$

and let's consider $O(f(n))$. It is a cluster and consists of all the low order function of f(n) and multiplies of f(n) such as 1,2,3, $cloglogn\ and\ clogn$ and

$O(g(n))$ is another cluster and consists of all the low order of n and multiplies of n Such as 1,2,3, $cloglogn , clogn, and\ cn$ so we can see this property holds

← second

as we look at reverse side

$this\ time\ let\ f(n) = 5^n\ and\ g(n) = logn * 7^n$

$so\ O(f(n))$ is 1,2,3,$loglogn , logn$,n, $n^2, \ldots 5^n\ and$

O(g(n)) is 1,2,3,$loglogn , logn$,n, $n^2, \ldots 5^n \ldots . 7^n \ldots logn * 7^n\ so\ \ O(f(n)) \subset O(g(n))$

And let's try to write $f(n) \in o(g(n))$ then

$5^n < clogn * 7^n$

exist positive constants c and n0 such as c =1 n0=2 n>n0

Consequently, this property holds

h) $f(n) \in \sim(g(n)) \Rightarrow f(n) \in \Theta(g(n))$

$$let\ f(n) = n * 5^n\ and\ g(n) = n * 5^n\ \ then\ we\ can\ write\ \ f(n) \in \sim(g(n))$$

Let’s try to write $f(n) \in \Theta(g(n))$

Then $c1 * g(n) \le f(n) \le c2 * g(n)$

$c1 * n * 5^n \le f(n) \le c2 * n * 5^n$ so, we can write c1 =1, c2=2 n0=1 n>=n0

Consequently, this property holds

i) $O(f(n)) = O(g(n)) \Leftrightarrow \Theta(f(n)) = \Theta(g(n)) \Leftrightarrow \Omega(f(n)) = \Omega(g(n))$

$$let\ f(n) = n!\ \ and\ \ g(n) = 7n! + na^n$$

$O(f(n))$ consists of 1,2,$loglogn$, $logn, n, n^2, n^3, n * a^n, cn!$
O(g(n)) consists of 1,2,$loglogn$, $logn, n, n^2, n^3, n * a^n, cn!$

So, first part holds

since na^n $has\ low\ order\ than$ $7n!$ $then$ don’t consider na^n
Θ ($f(n)$) consists of $1n!$,2 $n!$,$3n!$,$4n!$,$5n!$,$5n!$,$6n!$,$7n!$ ... .
Θ (g(n)) consists of $1n!$,2 $n!$,$3n!$,$4n!$,$5n!$,$6n!$,$7n!$ ... .
So, second part holds

Ω ($f(n)$) consists of $cn!, c2^{n^2}, c3^{n^2}, c4^{n^4}$ ....
Ω (g(n)) consists of $cn!, c2^{n^2}, c3^{n^2}, c4^{n^4}$ ....
So, third part holds

Questions – 3

a) $T(n)=\sum_{i=1}^{n-1} T(n)$+1   T (1) =1 then,

T (2) =T (1) +1=2

T (3) =T (2) +T (1) +1=4

T (4) =T (3) +T (2) +T (1) + 1=8

T (5) =T (4) +T (3) +T (2) + T (1) +1=16

T(n)=T(n-1) + T(n-2) …. T(1)+1=?

As we can see the relation clearly when n is 2 result is 2 when n is 3 result is 4 then when n is n result would be 2^{n-1} so

T(n)= 2^{n-1}

b) T(n)=T(n-2) + 3n+4, n>=3

T (1) =1, T (2) =6 then,

We need to consider 2 cases first one if the n is odd second one if the n is even However, in any case the equation looks like T(n)=T(n-8) +3(n-6) +3(n-4) +3(n-2) +3(n)+4+4+4 +4 in fourth step.

So, Let's take T (9) =T (1) + 3 (3) + 3 (5) + 3 (7) + 3 (9) +4+4+4+4

T (9) =T (1) + 3(3+5+7+9) +4(1+1+1+1) so we can see the general formula
T(n) = T (1) +3(3+5+7…n) + 4$\frac{(n-1)}{2}$ and it say us that
T(n)= 1+3($\frac{(n+1)^2}{4}$-1) + 4$\frac{(n-1)}{2}$

If n is even then again T(n)=T(n-6)+3(n-4)+3(n-2)+3(n)+4+4+4 such a equations in third step
And let's consider T(8)=T(2)+3(4)+3(6)+3(8)+4+4+4
So we have T(8)=T(2) + 3($\frac{(8+6+4)}{4}$ -2)+4(1+1+1) so it says us that the general formula

T(n)=T(2)+3($\frac{(\mathrm{n+n-2+n-4...})}{4}$ -2) + 4 ($\frac{(\mathrm{n-2})}{2}$)

T(n)=T(2)+3($\frac{\mathrm{n(n+2)}}{4}$ -2) + 4 ($\frac{(\mathrm{n-2})}{2}$)

c) x(n)=x($n^{\frac{1}{2}}$)+1

so when we step one more we get

x ($n^{\frac{1}{2}}$)= x($n^{\frac{1}{4}}$)+1 so main equation is x (n)=x($n^{\frac{1}{4}}$)+1+1

x ($n^{\frac{1}{4}}$)= x($n^{\frac{1}{8}}$)+1 so main equation is x (n)=x($n^{\frac{1}{8}}$)+1+1+1

.

.

So we can see that

x(2)=1

x(2^2)=x(2)+1=2

x(2^4)=x(2)+1+1=3

x(2^8)=x(2)+1+1+1=4 so when n's power is 2 we have one +1, when n's power is 4 we have two +1 so it says that the number of +1 is log ($log(n)$) and x(2) is always there.

Consequently, the formula is x(n)=1+ log ($log(n)$)

d) i) Disprove,

$$\text{let } f(n)=\frac{1}{n} \text{ so } f(n)^2 = \frac{1}{n^2}$$

f(n) has bigger order than $[f(n)]^2$

$$\lim_{n\to\infty} \left(\frac{\left(\frac{1}{n^2}\right)}{\frac{1}{n}}\right) = 0 \quad f(n) \notin O([f(n)]^2)$$

ii) Disproved,

f(n)=n^2 so one item of o(f(n)) is n^3 so

f(n) ∪ o(f(n)) ∉ Θ($f(n)$)

iii) Proved

$$f1(n) \leq c2*g1(n)$$

$$f2(n) \leq c4*g2(n) \text{ so}$$

$$f1(n)* f2(n) \leq c2*c4*g1(n)* g2(n)$$

Consequently

$$f1(n) * f2(n) \in O(g1(n) * g2(n))$$

iv) Proved

$$f1(n) \le c1g1(n) \quad , \qquad f2(n) \le c4g2(n),$$
$$f1(n) + f2(n) \le c8g1(n) + c9g2(n)$$
$$f1(n) + f2(n) \le cn1 * Max(g1, g2(n)) + cn2 * Max(g1, g2(n))$$
$$f1(n) + f2(n) \le (cn1 + cn2) * Max(g1, g2(n))$$

$$so\ it\ is\ proved$$

v) Disproved

$f1(n) = n$ and $f2(n) = n^2$

so, we can write

$$n \le c * g1(n) \quad and \quad n^2 \le c * g2(n)$$

Let $g1(n) = n\ \ and\ \ g2(n) = n^2 \qquad then\ we\ can\ find\ positive\ c\ constant$

So, min(g1(n), g2(n)) =n then

But we can write

$n + n^2 \nleq$ c.n

Consequently, it is disproved

vi) Disproved

Let f(n) return Boolean type like 1 or 0 if the parameter n is odd and zero if it is not, and g(n) = return 1 if n is even and zero otherwise.

Assume that f is member of O(g) we would say there is a constant C > 0 and N > 0 such that n > N implies f(n) <= C g(n). Let n is odd then we can say that n = 2 * N + 1. Then f(n) = 1 but g(n) = 0 so we cannot write f(n) <= C * g(n) since there is no c to make 0 bigger than any positive value. Thus, f is O(g) is not true.

e) $f(n) = 1 + k + k^2 + k^3 + \cdots .. + k^n$

the equation is f(n) = $\frac{1-k^{n+1}}{1-k}$ and if take its limit as n goes to infinity and k is less than 1 then it becomes f(n)= $\frac{1}{n-1}$ if k > 1 then it stay as it is.

then consider cases separately

if k<1 then the equation will be f(n)= $\frac{1}{n-1}$

so, we can say that $f(n) \in \Theta\left(\frac{1}{n-1}\right)$

if k=1 then the equation will be f(n)=n

so, we can say that $f(n) \in \Theta(n)$

if k>1 then the equation will be f(n) = $\frac{1-k^{n+1}}{1-k}$

so, we can say that 1-k is constant so $f(n) \in \Theta(1-k^{n+1})$

Questions – 4

Let's consider f(x)=$3(x-3)^2+4x+5\ln(x)$. Clearly f(x)∈ θ (n^2) but let's show it and low order and constant does not matter.

$$f(x) = 3(x^2-6x+9)+4x+5\ln(x)$$
$$f(x) = 3x^2-18x+27+4x+5\ln(x)$$
$$f(x) = 3x^2-14x+5\ln(x)+27$$

We see that f(x)∈ θ (n^2) let's write:

$$c1.n^2 \leq f(x) \leq c2.n^2$$

We can write these equations

$$14x \leq 3x^2\ asx \geq 5$$
$$5\ln(x) \leq 3x^2\ as\ x \geq 1$$
$$27 \leq 3x^2\ as\ x \geq 3$$

So, consider x>=5 and we can write our function

$$c1.n^2 \leq 3x^2-14x+5\ln(x)+27 \leq 3x^2+3x^2+3x^2+3x^2$$
$$c1.n^2 \leq 3x^2-14x+5\ln(x)+27 \leq 12x^2$$

$$c1.n^2 \leq f(x) \leq 12x^2$$

So low orders have no influence on complexity

And as for constant, if we have $400x^2\ instead\ 3x^2\ we\ still\ construct$ θ (n^2) and it will look like

$$c1.n^2 \leq f(x) \leq 1200x^2$$

So constant have no influence on complexity

Let's prove our theorem with graph and samples.

When we write such a program in python

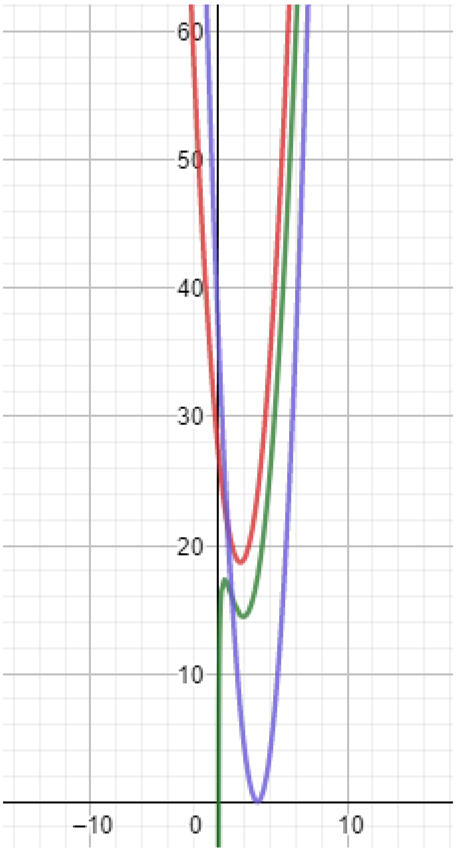
```
import math
n = list(range(5, 17))
f = [int(3*(x-3)*(x-3)+4*x+5*math.log(x, 2)) for x in n]
g = [3*(x-3)*(x-3)+8*x for x in n]
h = [4*(x-3)*(x-3) for x in n]


print(f)
print(g)
print(h)
```

The program will print out these lists

```
[43, 63, 90, 122, 159, 203, 253, 308, 370, 438, 511, 591]
[52, 75, 104, 139, 180, 227, 280, 339, 404, 475, 552, 635]
[16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576, 676]
```

And we can see that the results are really close



Here we have 3 function and these three functions grows the same after a certain x value since all these functions have $multiple\ of\ x^2$