17 - Apr - 2018 Complexity of an Algorithm.

$O(n^2 + 3n + 1) = n^2$. what does it mean?
$\exists$ Fixed n_0 and fixed ~~a~~ constant c ($\in \mathbb{N}$) s.t

$$|n^2 + 3n + 1| \leq cn^2, \quad \forall n \geq n_0 \qquad n \in \mathbb{N}^*$$

Eg: $O(\sqrt{n} + n) = n$.
$\exists$ n_0 and c s.t $|\sqrt{n} + n| \leq cn \ \forall n \geq n_0$

Definition: Polynomials, $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$
where $a_0, a_1, \dots a_n$ are $\mathbb{R}$
and all $n \in \mathbb{N}$.

Eg: $2x^3 + 7x + 2$ is a polynomial of degree 3.
$\frac{1}{2x^3 + 2x}$ is not a polynomial.

Fact: $O(\text{polynomial}) = n^{\text{highest exponent}}$
$= n^{\text{degree}}$

Eg. $O(2x^3 + 10x^5 - 7x + 10) = x^5$

Definitions: Lets call $2x^{3/2} + x^{1/2} + x + 2$ mini-polynomials
i.e looks like polynomial but all exponents are positive rational numbers.

Fact: $O(\text{mini polynomial}) = n^{\text{highest exponent}}$
$= n^{\text{degree}}$

Eg: $O(2n^{3/2} + n^{1/2} + n + 2) = n^{3/2}$

19-Apr-2018: Observe: eg: $n^2 + 3n + 7 \leq n^2 + 3n^2 + 7n^2$
$\leq 11n^2 \quad \forall n \in \mathbb{N}$.

Facts: $O(f_1 + f_2) = \max\{O(f_1), O(f_2)\}$
$O(f_1 \cdot f_2) = O(f_1) \cdot O(f_2)$.
$O\left(\frac{f_1}{f_2}\right) = \frac{O(f_1)}{O(f_2)}$

mini-function = $\frac{\text{mini polynomial}}{\text{mini polynomial}}$

rational function = $\frac{\text{polynomial}}{\text{polynomial}}$.

Hence $O(\text{mini-function}) = \frac{O(\text{mini polynomial})}{O(\text{mini polynomial})}$

and $O(\text{rational}) = \frac{O(\text{polynomial})}{O(\text{polynomial})}$.

eg: $O\left(\frac{\sqrt[4]{n^2} + \sqrt{n} + 3n}{n^{1/3} + n^{1/5} - 2}\right) = \frac{O(\sqrt[4]{n^2} + \sqrt{n} + 3n)}{O(n^{1/3} + n^{1/5} - 2)} = \frac{n}{n^{1/3}} = n^{2/3}$

Example:

```
For i = 2 to (3n+1)
    x = a*b + 1
    For k = 1 to i
    y = x ÷ 3 + b^2 - 1
    next k
next i.
```

i) Find the exact number of computation that is executed by the code?
ii) Find complexity of code.

Outer loop will be executed (terminal − initial + 1) times

i.e $(3n+1) - 2 + 1$

$= 3n$ times

let L be number of times outer loop executed.

$L: 1, 2, \cdots, 3n$ times

when $L = 1$, $i = 2$

no. of operations in outer loop = 2.

no. of iterations in inner loop * no. of op. in inner loop = $2 * 4$.

$\therefore$ total no. of op. at $L=1$; $\Rightarrow 2 + 2(4). = 10$

total no. of op. at $L = 2 \Rightarrow 2 + 3(4). = 14$

$\vdots$

total no. op. at $L = 3n \Rightarrow 2 + (3n+1)(4)$

$= 2 + 12n + 4$

$= 12n + 6.$

$\therefore$ total. no. of op. executed $= \sum_{L=1}^{3n}$ total no. op at L

$= \dfrac{(12n + 6 + 10)\,3n}{2}$

$= \dfrac{36n^2 + 48n}{2} = 18n^2 + 24n.$

ii) complexity $= n^2$

L : 1 $\qquad\qquad$ 3n

outer loop = 2 op.

$\rightarrow 2 + (2)(4) \qquad\qquad 2 + (3n+1)(4)$

$\Sigma = \dfrac{(10 + 2 + 12n + 4)}{2}\,3n$

$= (8 + 6n)\,3n$

$= 18n^2 + 24n.$

22 - Apr - 2018

Recall : $\lceil 2.5 \rceil = 3$
$\lceil -0.7 \rceil = 0$
$\lceil 3 \rceil = 3$ → ceiling function.

Definition : $\lceil x \rceil$, $x \in \mathbb{R}$ $= x$ if x is an integer
$\lceil x \rceil = y$ where y is the next integer after x, $x \neq$ integer

eg $\lceil 4.003 \rceil = 5$
$\lceil -3.002 \rceil = -3$

Floor function : $\lfloor x \rfloor = x$ if $x \in \mathbb{Z}$.
~~$\lfloor x \rfloor$~~ $\lfloor x \rfloor = y$, where y is the integer before x, $x \notin \mathbb{Z}$.

Eg: $\lfloor -3.42 \rfloor = -4$
$\lfloor 1.32 \rfloor = 1$
$\lfloor 7 \rfloor = 7$.

Assume $n \in \mathbb{N}^*$.
if n is odd, $\lceil \frac{n}{2} \rceil = \frac{n+1}{2}$ if n is odd, $\lfloor \frac{n}{2} \rfloor = \frac{n-1}{2}$

if n is even $\lceil \frac{n}{2} \rceil = \frac{n}{2}$ if n is even $\lfloor \frac{n}{2} \rfloor = \frac{n}{2}$.

```
For i=3 to ⌈n/2⌉
   x = 3*2*i
   For k=i to 8
   y = x*4*7÷3-x+8
   next k
next i.
```

i) Find the exact no. of multiplication, addition, subtraction, division that the code executes.
ii) Find complexity of the code.

Ans: i) # of times outer loop executed =
if n is even, $\lceil \frac{n}{2} \rceil = \frac{n}{2}$, $\frac{n}{2} - 3 + 1$ times i.e $\frac{n-4}{2}$ times
if n is odd, $\lceil \frac{n}{2} \rceil = \frac{n+1}{2}$, $\frac{n+1}{2} - 3 + 1$ times i.e $\frac{n-3}{2}$ times.
# of operations in outer loop = 2
# of operations in ~~the~~ inner loop = 5

Outer loop # of iterations : $1, 2, \ldots \frac{n-4}{2}$ n is even.

At iteration 1 : Total no. of op = $2 + 6\times5$ i.e $2 + (8-3+1)\times5$
iteration 2 : Total no. of op = $2 + 5\times5$
" 3 : Total no. of op = $2 + 4\times5$
⋮
" $\frac{n-4}{2}$: Total no. of op = $2 + \left(8 - \frac{n}{2} + 1\right)\times5$
$= 2 + \left(\frac{18-n}{2}\right)\times5$

Forms arithmetic sequence: $2+5(6), 2+5(5), 2+5(4), \ldots, 2+5\left(\frac{18-n}{2}\right)$
difference = 5

Σ sequence $= \frac{(a_1 + a_m)m}{2}$

$= \frac{\left[32 + 2 + 5\left(\frac{18-n}{2}\right)\right]\left(\frac{n-4}{2}\right)}{2}$

$= \frac{\left(32 + 2 + 45 - \frac{5n}{2}\right)\frac{n-4}{2}}{2}$

$= \frac{\left(79 - \frac{5}{2}n\right)\frac{n-4}{2}}{2}$

$\therefore$ highest power will be n^2

In n is odd:

L: $1, 2, \ldots, \frac{n-3}{2}$

Total no. of operations in each iteration : $2+5(6)$, $2+5(5)$, ..., $2+5\left(8-\frac{(n+1)}{2}+1\right)$

$2+5 \cancel{\frac{19-n}{2}} \left(\frac{17-n}{2}\right)$

$\Sigma = \frac{32 + 2 + 5\left(\frac{17-n}{2}\right)}{2} \times \frac{n-3}{2}$

$\therefore$ highest power will be n^2.

ii) Complexity of code $= n^2$.