FOUNDATIONS OF ALGORITHMS - CSE 551 - ASSIGNMENT 1

1.

(i) $n! \in O(n^n)$ - ***True***

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

In this case: f(n) = n! And $g(n) = n^n$
From the above definition,

$$0<=f(n)<=cg(n) \text{ for all } n>=k$$
$$0<=n!<=cn^n \text{ for all } n>=k$$

Considering k=1 and n=1 gives us c>=1
Considering c=1:

$$0<=n!<=n^n$$

Since n! Is greater than 0 for all n>=0, the left inequality is satisfied automatically.

$$n! <=n^n$$
$$nx(n-1)!<=nxn^{n-1}$$
$$(n-1)!<=n^{n-1}$$

For n=2 : 1<=2 (True)
For n=3 : 2<==9 (True)
For n=7 : 720<=117649 (True)

Hence, for any value of n, the above conditions hold true and therefore $n! \in O(n^n)$ is true.

(ii) $2n^22^n + n\log n \in \Theta(n^22^n)$ - ***True***

For a given function g(n), Θ(g(n)) is given by:
Θ(g(n)) = { f(n) : there exists positive constants c_1, c_2 and k such that 0 <= c_1g(n) <= f(n) <= c_2g(n) for all n>=k}

In this case, $f(n)=2n^22^n + n\log n$ and $g(n)=n^22^n$
From the above definition,

$$c_1n^22^n <= 2n^22^n + n\log n <= c_2n^22^n \text{ for all } n>=k$$
$$c_1 <= 2 + (\log n/n2^n) <= c_2$$

Considering k=1 and n=1 gives us $c_1<=2$
Considering k=2 and n=2 gives us $c_2>=2.0375$

Considering c_1=2 and c_2=2.0375:

$$2 <= 2 + (logn/n2^n) <= 2.0375$$
$$0 <= logn/n2^n <= 0.0375$$

For n=1 : 0<=0<=0.0375 (True)
For n=2 : 0<=0.0375<=0.0375 (True)
For n=3 : 0<=0.01988<=0.0375 (True)
For n=4 : 0<=0.0094<=0.0375 (True)

As seen above, the inequality holds for all the values of n (As n tends to infinity, $logn/n2^n$ tends to 0) and hence, $2n^2 2^n + nlogn \in \Theta(n^2 2^n)$ is true.

(iii) $10n^2+9 = O(n)$ - ***False***

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

In this case, $f(n)=10n^2+9$ and g(n)=n
From the above definition,

$$0<=10n^2+9<=cn \text{ for all } n>=k$$

Considering k=1 and n=1 gives us c>=19
Considering c=19:

$$0<=10n^2+9<=19n$$

Since n^2 is greater than or equal to 0, $10n^2+9$ is always greater than 0 and that makes the left inequality to be true.

$$10n^2+9<=19n$$
$$10n^2-19n+9<=0$$

$$10n^2-10n-9n+9<=0$$
$$10n(n-1)+9(n-1)<=0$$
$$(10n-9)(n-1)<=0$$
$$0.9<=n<=1$$

So, whenever n lies between 0.9 and 1, the condition holds true and for any other value of n, the condition does not work. Hence, $10n^2+9 = O(n)$ is false.

(iv) $n^2logn = \Theta(n^2)$ - ***False***

For a given function g(n), Θ(g(n)) is given by:
Θ(g(n)) = { f(n) : there exists positive constants c_1, c_2 and k such that 0 <= c_1g(n) <= f(n) <= c_2g(n) for all n>=k}

In this case, $f(n)=n^2logn$ and $g(n)=n^2$

From the above definition,

$$c_1n^2 <= n^2logn <= c_2n^2 \text{ for all } n>=k$$

$$c_1 <= logn <= c_2$$

Considering k=1 and n=1 gives us $c_1<=0$
Considering k=10 and n=10 gives us $c_2>=1$

Considering $c_1=0$ and $c_2=1$:

$$0<=logn<=1$$

For n=1: 0<=0<=1 (True)
For n=10 : 0<=1<=1 (True)
For n=100 : 0<=2<=1 (False)
For n=1000 : 0<=3<=1 (False)

As seen above, the inequality does not work for larger values of n and hence, $n^2logn = \Theta(n^2)$ is false.

(v) $n^32^n + 6n^23^n = O(n^32^n)$ - ***False***

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

In this case, $f(n)=n^32^n + 6n^23^n$ and $g(n)=n^32^n$
From the above definition,

$$0<=n^32^n + 6n^23^n<=cn^32^n \text{ for all } n>=k$$

Considering k=1 and n=1 gives us c>=10
Considering c=10:

$$0<=n^32^n + 6n^23^n<=10n^32^n$$

Since $n^2,n^3,2^n,3^n$ are all greater than or equal to 0 for all n>=0, $n^32^n + 6n^23^n$ is always greater than 0 and that makes the left inequality to be true.

$$n^32^n + 6n^23^n<=10n^32^n$$

$$9n^32^n>=6n^23^n$$

$$nx2^{n-1}>=3^{n-1}$$

For n=2 : 4>=3 (True)
For n=5 : 80>=243 (False)
For n=10 : 5120>=19683 (False)

As seen above, the inequality does not work for larger values of n and hence, $n^32^n + 6n^23^n = O(n^32^n)$ is false.

2.

The computer performs 10^{10} operations per second. So, for one hour of computation, the number of operations that will be performed by the computer are $60 \times 60 \times 10^{10} = 36 \times 10^{12}$.

(i) Given algorithm's running time: n^2.
The largest input size n for which the above computer will be able to get the result in one hour:

$$n^2 = 36 \times 10^{12}$$
$$n = 6 \times 10^6$$

Hence, the largest input size that can be computed in one hour = n = ***6×10^6***

(ii) Given algorithm's running time: n^3.
The largest input size n for which the above computer will be able to get the result in one hour:

$$n^3 = 36 \times 10^{12}$$
$$n = 3.3019 \times 10^4 = 33019$$

Hence, the largest input size that can be computed in one hour = n = ***33019***

(iii) Given algorithm's running time: $50n^2$.
The largest input size n for which the above computer will be able to get the result in one hour:

$$50n^2 = 36 \times 10^{12}$$
$$n = 0.848528 \times 10^6 = 848528$$

Hence, the largest input size that can be computed in one hour = n = ***848528***

(iv) Given algorithm's running time: 3^n.
The largest input size n for which the above computer will be able to get the result in one hour:

$$3^n = 36 \times 10^{12}$$
$$n = \log_3 36 + 12 \times \log_3 10$$
$$n = 3.2618 + 12 \times 2.0959 = 28.4126 \sim 28$$

Hence, the largest input size that can be computed in one hour = n = ***28***

3.

Algorithm A_1 takes $10^{-3} \times 2^n$ seconds for solving a problem instance of size n and Algorithm A_2 takes $10^{-2} \times n^4$ to solve the same on a particular machine.

(i) Number of seconds in one year = 365 x 24 x 60 x 60 = 31536000
The size of the largest problem instance A_2 will be able to solve in one year:

$$10^{-2} \times n^4 = 31536000$$
$$n^4 = 315360 \times 10^4$$
$$n = 23.6974 \times 10 = 236.974 \sim 236$$

Hence, the largest problem instance A_2 will be able to solve in the given amount of time = n = ***236***

(ii) Number of seconds in one year = 365 x 24 x 60 x 60 = 31536000
The size of the largest problem instance A_2 will be able to solve in one year on a machine hundred times faster than the previous mentioned machine:

$$10^{-2} \times n^4 = 31536000 \times 100$$
$$n^4 = 31536000 \times 10^4$$
$$n = 74.9379 \times 10 = 749.379 \sim 749$$

Hence, the largest problem instance A_2 will be able to solve in the given amount of time on a hundred times faster machine = n = ***749***

(iii)
Assuming n = 19:
The time algorithm A_1 takes to solve an instance of size 19 = $10^{-3} \times 2^{19}$ = 524.288 seconds
The time algorithm A_2 takes to solve an instance of size 19 = $10^{-2} \times 19^4$ = 1303.21 seconds
So, A_1 takes less time to solve an instance of size 19 when compared to A_2

Assuming n = 20:
The time algorithm A_1 takes to solve an instance of size 20 = $10^{-3} \times 2^{20}$ = 1048.576 seconds
The time algorithm A_2 takes to solve an instance of size 20 = $10^{-2} \times 20^4$ = 1600 seconds
So, A_1 takes less time to solve an instance of size 20 when compared to A_2

Now, assuming n = 21:
The time algorithm A_1 takes to solve an instance of size 21 = $10^{-3} \times 2^{21}$ = 2097.152 seconds
The time algorithm A_2 takes to solve an instance of size 21 = $10^{-2} \times 21^4$ = 1944.81 seconds
So, A_2 takes less time to solve an instance of size 21 when compared to A_1

Hence, ***the algorithm A_1 is faster than the algorithm A_2 when the size of the instance is less than or equal to 20*** but when the size of the instance is greater than 20, the algorithm A_2 is faster than A_1.

4.

The ascending order of the growth rate for the given functions: ***(ii) < (iii) < (i) < (iv) < (v)***

Proof:

$f_1(n) = n^{4.5} = 3^{4.5}$
$f_2(n) = (3n)^{0.6}$
$f_3(n) = n^4 + 20$
$f_4(n) = 25^n$
$f_5(n) = 260^n$

Considering the part (ii)<(iii):

$$(3n)^{0.6} <= n^4 + 20$$

$$(3n)^{0.6} \epsilon O(n^4 + 20)$$

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

$$0<=(3n)^{0.6} <= c.(n^4 + 20) \text{ for all } n>k$$

Assuming k=1 and n=1 gives us c >= 0.092
Considering c = 1:

$$0<=(3n)^{0.6} <= n^4 + 20$$

The left inequality holds true for all n>0.

$$(3n)^{0.6} <= n^4 + 20$$

For n=2 : 2.93 <= 36 (True)
For n=3 : 3.73 <= 101 (True)

Hence, for any value of n, the above condition stays true and hence $(3n)^{0.6} <= n^4 + 20$ is true.

Considering the part (iii)<(i):

$$n^4 + 20 <= n^{4.5}$$

$$n^4 + 20 \epsilon O(n^{4.5})$$

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

$$0<=n^4 + 20<=c.n^{4.5} \text{ for all } n>k$$

Assuming k=1 and n=1 gives us c >= 21
Considering c = 21:

$$0<=n^4 + 20<=21n^{4.5}$$

The left inequality holds true for all n>0.

$$n^4 + 20<=21n^{4.5}$$

For n=2 : 36 <= 475.17 (True)
For n=3 : 101 <= 2946.21 (True)

Hence, for any value of n, the above condition stays true and hence $n^4 + 20 <= n^{4.5}$ is true.

Considering the part (i)<(iv):

$$n^{4.5} <= 25^n$$

$$n^{4.5} \in O(25^n)$$

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

$$0<=n^{4.5}<=c.25^n \text{ for all } n>k$$

Assuming k=1 and n=1 gives us c >= 0.04
Considering c = 0.04:

$$0<=n^{4.5}<=0.04x25^n$$

The left inequality holds true for all n>0.

$$n^{4.5}<=0.04x25^n$$

For n=2 : 22.62 <= 25 (True)
For n=3 : 140.29 <= 625 (True)

Hence, for any value of n, the above condition stays true and hence $n^{4.5} <= 25^n$ is true.

Considering the part (iv)<(v):

$$25^n <= 260^n$$

$25^n \in O(260^n)$

For a given function g(n), O(g(n)) is given by:
O(g(n)) = { f(n) : there exists positive constants c and k such that 0 <= f(n) <= cg(n) for all n>=k}

$0 <= 25^n <= c.260^n$ for all n>k

Assuming k=1 and n=1 gives us c >= 0.096
Considering c = 0.096:

$0 <= 25^n <= 0.096x260^n$

The left inequality holds true for all n>0.

$25^n <= 0.096x260^n$

For n=2 : 625 <= 6489.6 (True)

For n=3 : 15625 <= 1687296 (True)

Hence, for any value of n, the above condition stays true and hence $n^{4.5} <= 25^n$ is true.

Therefore, inferring from the proof of the above four parts, the ascending order of the growth rate for the given functions is (ii) < (iii) < (i) < (iv) < (v)