Ques1 $O(n^2)$ for $n = 10 \rightarrow 5$ sec

$100 \rightarrow 5$

$2500 \rightarrow \frac{5}{100} \times 2500 = 125$ sec

Ques2

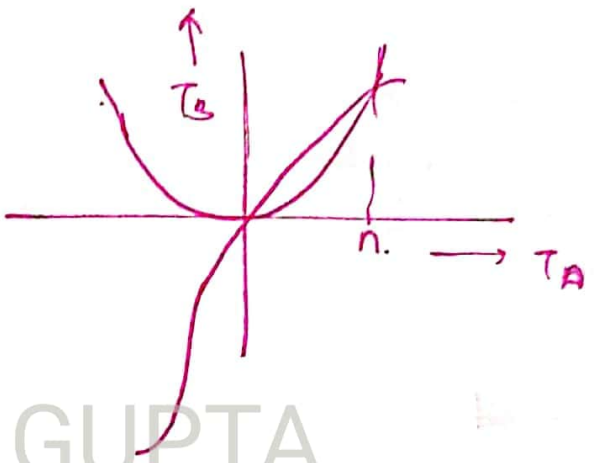
$T_A(n) = n^3$

$T_B(n) = 2n^2$

$n^3 - 2n^2 = 0$

$n^2(n-2) = 0$

$\boxed{n = 2}$



Ques3 $f(n) = n2^n$ $O(4^n)$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \frac{n2^n}{4^n} = n\left(\frac{1}{2}\right)^n$$

$$\lim_{n \to \infty} \frac{n}{2^n} \Rightarrow \lim_{n \to \infty} = 0$$

$O(f(n)) < O(g(n))$

so, the $n2^n$ is in $O(4^n)$.

Ques4 $O(\log_n n)$

An algorithem's runing time depends on its input sizes

ex, $\log_3 27$ is 3

logarithm function gets slightly slower as n grows where n doubles the running time lowered by constant

Ques.5

(a) Worst case (O) is the func" which performs the maximum numbers of steps on input data of size n.
ex - if we want to count an array of integers in asending order then the worst case would be that array is already sroted in desending order.

Average case (Θ) is used when we refer to average no. of input to test the algorithem. It is the average resources used taking into all possible inputs.

(b) Big-oh notation is an asympote notation for the asympatic upper bound for the growth rate of the running of an algorithm. (O)

Ω - Asymptotic notation for best case it provides us with an asymptotic lower bond for the growth rate of the running algorithm.

⑥ $n^4 + \log(n) + 17$

growth rate of constant —

$O(n^4) > O(\log n) > O(1)$

So, $n^4 + \log n$

$= n^4\ O(n^4)$

As we consider the input size & increase it the growth rate of logarithm function is lesser than the polynomial function.

⑦

```
k = 1                    ⟶ O(1)
while (k ≤ n)            ⟶ O(n)
{
    k = k+1;             ⟶ O(n)
}
```

N = 5

```
for (i = 1 to n-1)       ⟶ O(n)
{
    for (j = i+1 to n)   ⟶ O(n-1)
    {
        swap             ⟶ O(n-1)
    }
}
```


Time complexity ⇒ $O(n) + O(n-1) + O(n-1)$

$= O(n)$

⑧ $f(n) = n^2 + 2n + 4$

$O(n^2) > O(n) > O(1)$

$f(n) = O(n^2)$

As increase the input size double then the running time will become larger

As the graph show as the input increases time also increases.

Expt. No. _____ Page No. _____

(9) $T_A = \log n$ $T_B = n$

$$\lim_{n \to \infty} \frac{T_A}{T_B} = \frac{\log n}{n^4}$$

$T_B(n) < T_A(n)$

$n^4 < \log n$

$$\lim_{n \to \infty} \frac{\log n}{n^4} = 0$$

Thus, $f(n)$ is in $O(g(n))$ thus n^4 is a upper bound of $(100)^n$.

as $n \to \infty$ T_A will grow faster then T_B function.

(10) $n\log n \in O(\log n!)$
as n increases

$\log(n!) = \log(1) + \log 2 + \dots \log(n-1) + \log(n)$

$n \log n \to \log 1 + 2\log 2 \dots n \log n$

(11) $2^{n-1} + 4^{n-1}$

$T_A = n^2$

$T_B = n+2$

$n^2 = n+2$

$n^2 - n - 2 = 0$

$\boxed{n=2}$ → Breaking point

Teacher's Signature _____

(14)

```
total = 0                    → O(1)
for (i = 0 to n-1)           → O(n-1)
{
sum = sum + a[i]             → c (O(n))
  return sum                 → c(O(n))
}
```

$T_{sum} = O(1) + O(n-1) + c(O(n-1) + \ldots$

$\rightarrow O(n)$

(15)

```
int count = 0;               → O(1)
for (i = 0; i < N; i++)      → O(n)
{
  for (j = 0; j < i; j++)
  {
    count++                  → c(O(n))
  }
```