DSA 90 days challenge Day 1

Time Complexity

Time complexity is a type of compulational complexity that describe the time required to execute an algorithm.

The time complexity of an algorithm is the amount of time it takes for each statement to complete. As a result, it is highly dependent on the size of the processed data.

Algorithm

1.
```
A(n)
{
  int i,
  for(i=1 to n)
    Pf("Rawi") → n
}
```
O(n)

2.
```
A(n)
{
  int i, j;
for(i=1 to n)
for(i=1 to n)
    Pf(rawi) → n²
}
```
O(n²)

(3)
```
A(n)
{
  i=1, s=1;
while (s<=n)
{
  i++;
  s=s+i
  PF("rawi")
}
}
```
Sum of natural number

$\frac{k(k+1)}{2} > n$

$\frac{k^2+1}{2} > n$

$k = (\sqrt{n})$

④
```
A()
{
  i=1
for (i=1; i*i<=n; i++)
  PF ("ravi");  ⟶ √n
}
```

$O(\sqrt{n})$. Answer.

⑤
```
A()
{
  int i, j, k, n;
  for (i=1; i<=n; i++)
  {
    for (j=1; j<=i; j++)
    {
      for (k=1; k<=100; k++)
      {
        PF ("Ravi");
      }
    }
}
```

solve →

$i = 1$	$i = 2$	$i = 3$
$j = 1$ times	$j = 2$ times	$j = 3$ times
$K = 100$	$K = 2 * 100$	$K = 3 *$ times

$i = 4$	$i = 5$	.....	$j = n$
$j = 4$ times	$j = 5$ times		$j = n$ times
$K = 4 * 100$	$K = 5 * 100$		$K = n * 100$

we are going to analyse the above things

$100 + 2 \times 100 + 3 \times 100 + \cdots\cdots n \times 100$

$= 100(1 + 2 + 3 + \cdots\cdots n)$

$= 100 \dfrac{n(n+1)}{2}$

$O(n^2)$ Answer

6

```
A()
{
  int i, j, K, n

  for (i=1; i<=n; i++)
  {
    for (j=1; j<=i^2; j++)
    {
      for (K=1; K<=n/2; K++)
      {
        PF("Rani")
```

$i = 1$	$i = 2$	$i = 3$
$j = 1$ times	$j = 2^2 = 4$	$j = 3^2 = 9$
$K = \frac{n}{2} * 1$	$K = \frac{n}{2} * 4$	$K = \frac{n}{2} * 9$

$i = 5$
$j = 5^2 = 25$ times
$K = \frac{n}{2} * 25$

$$\frac{n}{2} * 1 + \frac{n}{2} * 4 + \frac{n}{2} * 9 \ldots\ldots \frac{n}{2} * n^2$$

$$\frac{n}{2}(1 + 4 + 9 + \text{-------} n^2)$$

This is nothing but sum of first n square of natural number

$$\frac{n}{2}\left(\frac{n(n+1)(2n+1)}{6}\right)$$

$O(n^4)$ Answer

(7)
```
A(n)
{
for (i=1; i<n; i*2)
   PF("raj");
}
```

log base depending on the multiple value.

$i = 1, 2, 4, 8, 16 \ldots\ldots (n)$
$2^0\ 2^1\ 2^2\ 2^3 \ldots 2^K$

$2^K = n$	$K \log 2 = \log n$
$\log 2^K = \log n$	$O(\log_2 n)$

8 A()

{

int i, j, k;

$n/2$ — for ($i = \frac{n}{2}$; $i <= n$; i++)

$n/2$ — for ($j = 1$; $i <= \frac{n}{2}$; j++)

$\log_2 n$ — for ($K = 1$; $K \leq n$; $K = K * 2$)

PF("Rani")

$\frac{n}{2} * \frac{n}{2} * \log_2 n$

$\boxed{O(n^2 \log_2 n)}$

9. A()

{

int i, j, K;

for ($i = n/2$; $i <= n$; i++) ⟶ $n/2$

for ($i = 1$; $j <= n$; $j = 2 * j$) ⟶ $\log_2 n$

for ($K = 1$, $K <= n$; $K = K * 2$) ⟶ $\log_2 n$

}

$n/2 (\log_2 n)^2$

$O(n(\log_2 n)^2$ Answer