

Dry run of Time, space Complexity

Problem 58-

1. Dry run the code for $n=4$. How many times is * printed? What is the Time Complexity?

```
void printTriangle (int n) {  
    for (int i = 0; i < n; i++)  
        for (int j = 0; j <= i; j++)  
            System.out.println ("*");  
}
```

→ Execution :-

i (outer loop)	j (inner loop)	Output
0	0	*
1	0, 1	* *
2	0, 1, 2	* * *
3	0, 1, 2, 3	* * * *

Total stars printed = 10. $1+2+3+4 = 10$

Time Complexity :-

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

$$= \frac{n^2 + n}{2}$$

$$\text{Time Complexity} = \underline{\underline{O(n^2)}}$$

2. Dry run for $n=8$. What's the number of iterations? Time Complexity?

```

void printPattern (int n) {
    for (int i=1; i<=n; i*=2)
        for (int j=0; j<n; j++)
            System.out.println (i+" "+j);
}

```



Execution :-

(Outer loop i) Iteration.	Value of i	(inner loop j) (0 to n-1)	Total Print Per i
1st	$i=1$	0, 1, 2, 3, 4, 5, 6, 7	8
2nd	$i=2$	0, 1, 2, 3, 4, 5, 6, 7	8
3rd	$i=4$	0, 1, 2, 3, 4, 5, 6, 7	8
4th	$i=8$	0, 1, 2, 3, 4, 5, 6, 7	8

$$\text{Total} = 8 + 8 + 8 + 8 = 32$$

Time Complexity :- $O(n \log n)$.

3. Dry run for $n = 20$. How many recursive calls? What values are printed?

```
void recHalf (int n) {  
    if (n <= 0) return;  
    System.out.println (n + " ");  
    recHalf (n/2);  
}
```

→

Recursive call	Value of n	Condition check ($n \leq 0$) = ?	Print
1 st call	20	$20 > 0 \Rightarrow \text{No.}$	20.
2 nd	10 ($20/2$)	$10 > 0 = \text{No}$	10
3 rd	5 ($10/2$)	$5 > 0$	5
4 th	2 ($5/2$)	$2 > 0$	2
5 th	1 ($2/2$)	$1 > 0$	1
6 th	0 ($1/2$)	$0 = 0 \Rightarrow \text{Return}$	-

Total ^{output} ~~calls~~ = 20, 10, 5, 2, 1.

Total call = 6.

Time complexity = $O(\log_2 n)$.

4. Dry run for $n=3$. How many total calls are made? What's the time complexity?

```
void fun(int n) {  
    if (n == 0) return;  
    fun(n-1);  
    fun(n-1);  
}
```

→	Value of n	no. of calls.
0	$n=3$	1
1	$n=2$	2
2	$n=1$	4
3	0	8 return

$$\text{Total} = 1 + 2 + 4 + 8 = 15$$

Time complexity =

2

5. Dry run for $n=3$. How many total iterations?
Time complexity?

```

void tripleNested (int n) {
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
            for (int k=0; k<n; k++)
                S.o.p (i+j+k);
}
    
```

→ Iteration:

outer loop (i)	middle loop (j)	inner (k)	Print (i+j+k)
0	0	0	0
0	0	1	1
0	0	2	2
0	1	0	1
0	1	1	2
0	1	2	3
0	2	0	2
0	2	1	3
0	2	2	4
1	0	0	1
1	0	1	2
1	0	2	3
1	1	0	2
1	1	1	3
1	1	2	4
1	2	0	3
1	2	1	4
1	2	2	5

i	j	k	pair (i+j+k)
0	0	0	2
2	0	1	3
2	0	2	4
2	0	2	3
2	1	0	4
2	1	1	5
2	1	2	4
2	2	0	5
2	2	1	5
2	2	2	6

iteration:- $n=3$.

$$3 \times 3 \times 3 = 27.$$

time complexity :- $O(n^3)$.