Name: Shriniwas Vasant Pawar

Center : Kharghar

PRN: 250240320110

Problem 5:

Dry Run & Analyze: Time and Space Complexity

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.print("*");
}
```

Ans: * will be printed 10 times, Time Complexity: $O(n^2)$, Space Complexity: O(1)

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);
}
```

Ans: There are 32 number of iterations. Time Complexity: O(nlogn), Space Complexity: O(1).

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

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

Ans: There are 5 recursive calls are there. Values printed: 20 10 5 2 1 . Time Complexity: O(logn), Space Complexity: O(log 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);
}
```

Ans: 6 total calls are made. Time Complexity: $O(2^n)$, Space Complexity: O(n)

---

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++)
System.out.println(i + j + k);
}
```

Ans: total iterations are 27 , Time Complexity: $O(n^3)$, Space Complexity: O(1).