

25. SWITCH CASE STATEMENT

**TO EXAMINE SWITCH CASE IS JUST LIKE
IF AND ELSE:**

a = 1, b = 2, c = 3

```
switch(a){  
    case 1: — — — — — — — —→ O(1)  
        print(Monday);  
        break;  
    case 2: — — — — — — — —→ O(1)  
        print(Tuesday);  
        break;  
    case 3: — — — — — — — —→ O(1)  
        print(Wednesday);  
        break;  
    default: — — — — — — — —→ O(1)  
        print("Not Found");  
        break;  
}
```

NOTE: every case runs a constant and returns $O(1)$ complexity.

Differentiation between Switch Case and If-else.

As switch takes $a = 1$, hence case A runs i. e. c_1

Which it creates differentiation between if else statement. As it takes a key in switch then it jumps to the key = case and run the case statement.

Or if case is not found i. e. key doesnot get matched with any cases of switch block then default statement is executed.

Single case run time is usually $O(1)$.

But there are some exceptions.

1. Switch run $O(n)$ times which case $O(1)$ times:

```
for (int i = 1; i ≤ n ; i ++){  
    switch(i){  
        case 1:  
            print("hello");
```

```

        break;
    case 2:
        print("hello");
        break;
    default:
        print("Not Found");
        break;
}

```

ANSWER

Here switch case take n no. inputs which cases takes 1 unit of time give switch case time complexity : $cn = O(n)$ if case is found or default runs it runs `n` times .

2. Switch run `O(n)` times which case O(n) times:

```

int n = 5;
int sum = 0;
for (int i = 1; i ≤ n ; i ++){
    switch (i) {
        case 1:

```

```

    for (int j = 1; j <= n; j++) {
        sum = sum + j;
    }
    break;
case 2:
    for (int j = 1; j <= n; j++) {
        sum = sum + j;
    }
    break;
// more cases here...
}

default:
    print("Execution Finished");
    break;
}
}

```

ANSWER

Here switch case take n no. inputs and cases takes n unit of time give switch case time complexity ($n \times n$) :

$$cn^2 = O(n^2).$$

default := $n - 1$ times.

at nth time := Case runs = n^2 , hence it will run

$$n - 1 + n^2 = O(n^2),$$

default is neglected ,when case is matched.

*And total complexity of for loop is $n \times n^2 = O(n^3)$.
if at n^{th} input case remains `not found` then ,
default gets executed `n` times gives
— $O(n)$ complexity.*

3. Switch run $O(n)$ times which case $O(\log n)$ times:

```
int n = 5;  
int sum = 0;  
for (int i = 1; i ≤ n ; i ++){  
    switch (i) {  
        case 1:  
            for (int j = 1; j ≤ n; j = j * 2) {  
                sum = sum + j;  
            }  
            break;  
        case 2:  
            for (int j = 1; j ≤ n; j = j * 2) {  
                sum = sum + j;  
            }  
            break;  
        // more cases here...  
    }
```

default:

```
print("Execution Finished");  
break;  
}  
}
```

ANSWER

Here switch case take n no. inputs and cases takes log n unit of time give switch case time complexity (nlogn) :

$$c(n \log n) = O(n \log n).$$

default runs := n - 1 times.

at nth time := Case runs = nlogn , hence it will run

$$\text{Therefore } T(n) = n - 1 + n \log n = O(n \log n),$$

default is neglected ,when case is matched.

*And total complexity of for loop is $n \times n \log n$
 $= O(n^2 \log n)$.*

*if at n^{th} input case remains `not found` then ,
default gets executed `n` times gives*

– $O(n)$ complexity
