25. SWITCH CASE STATEMENT

TO EXAMINE SWITCH CASE IS JUST LIKE IF AND ELSE:

```
$\boldsymbol{a = 1, b = 2, c = 3}$
$\boldsymbol{switch(a)}\{$
   $\boldsymbol{case\ 1}: -------\!-\!\!\rightarrow \boldsymbol{O(1)}$
      $\boldsymbol{print}(\text{Monday});$
       $\boldsymbol{break};$
  $\boldsymbol{case\ 2}: --------\!-\!\!\rightarrow \boldsymbol{O(1)}$
      $\boldsymbol{print}(\text{Tuesday});$
       $\boldsymbol{break};$
  $\boldsymbol{case\ 3}: -------\!\!\rightarrow \boldsymbol{O(1)}$
      $\boldsymbol{print}(\text{Wednesday});$
       $\boldsymbol{break};$
  $\boldsymbol{default}: -------\!\!\rightarrow \boldsymbol{O(1)}$
      $\boldsymbol{print}(\text{"Execution Finished"});$
       $\boldsymbol{break};$
$\}$
```

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.

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

ANSWER

Here switch case take n no. inputs which cases takes 1 unit of time give switch case time complexity :

$c_1 + c_2 n = O(n)$

as default always gets added at last , always runs constant time c.

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

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×n) :

$c_1 + c_2n^2 = O(n^2)$

as default always gets added at last , always runs constant time c.

And total complexity of for loop is $n \times n^2 = O(n^3)$.

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

$c_1 + c_2(\,\mathrm{nlog}\,n\,) = O(\mathrm{nlog}\,n)$

$as\ default\ always\ gets\ added\ at\ last\ ,always\ runs\ constant\ time\ c.$

$And\ total\ complexity\ of\ for\ loop\ is\ n \times \mathrm{nlog}\,n$

$$= O\left(n^2 logn\right).$$