Analysis of common Loops

Evergreen
Page No.
Date : / /20

1 * O(1)

a statement or part of code is said O(1) when it doesn't contain any
(i) recursive function or
(ii) iterative loop.

Eg :- Swap().

- also, when a loop runs for constant num of times is also said O(1)

Eg :-
```
for(int i=0; i<4; i++){
    System.out.Print("Hello");
}
```

here loop will run upto less than 4 means up to 3 only.
it's a constant so, O(1).

2 * O(n)

Time complexity of loop is consider O(n) when loop runs for constant amount up to the n. the last element of the input same in recursion.

Eg :-
```
for(int i=0; i<n; i++){
    Sout("Hello").
}
```

it'll run upto n num of time.

3 * $O(n^2)$

Time complexity is $O(n^2)$ when there is a inner or nested loop of equal num times. means both runs for same amount.

outer loop.

```
eg:- for (int i=0 ; i<n ; i++){
        for (int j=0 ; j<n ; j++){
        }
     }
```

(same amount (n).)

nested loop

∴ here both loops are running upto same amount "n".

so, $n \times n = n^2$.

(outer, nested or inner)

4 * $O(\log n)$

Time complexity of a loop is considered as $O(\log n)$ if the loop variables are devided / multiplied by a constant amount.

multiply constant amount of time.

```
Eg:- for (int i=1 ; i<n ; i = i * 3){
        cout ("Hello");
     }
```

dividing constant amount of time.

```
for (int i=1 ; i<n ; i = i/9){
        cout ("Hello");
}
```

i.e Binary search has time ~~complexity~~ $O(\log n)$