

Big O notation

Qn) linear time

```
int addUp(int n) {  
    int sum = 0;  
    for (int i = 0; i <= n; i++) {  
        sum += i;  
    }  
}
```

}

return sum;

}

$n = 1000000$

~ 1000000 steps

Qn) constant time

~~Qn~~

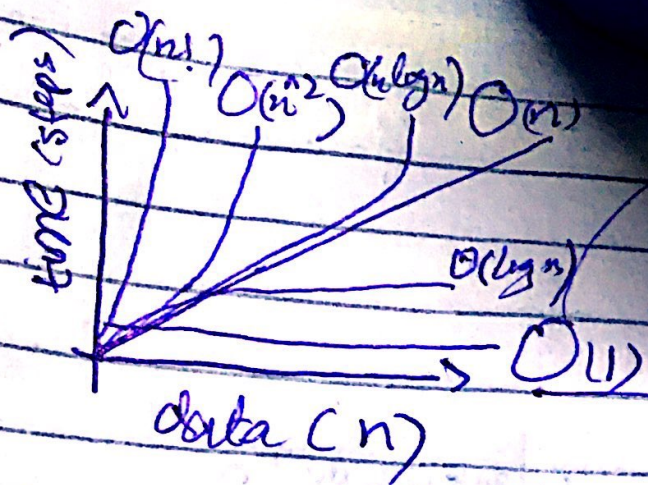
```
int addUp(int n) {
```

```
    int sum =  $n * (n+1) / 2$ ;  
    return sum;  
}
```

}

$n = 1000,000$
3 steps.

$O(3)$ but this small no. of steps don't matter so we say it is constant time
 $O(1)$



- $O(1)$ = constant time
- random access of an element in an array
 - inserting @ the beginning of linked list

$O(\log n)$ = logarithmic time

- binary search

$O(n)$ = linear time

- looping thru elements in an array
- searching thru a linked list

$O(n \log n)$ = quasilinear time

- quick sort, merge sort, heap sort

$O(n^2)$ = quadratic time

- insertion sort, selection sort, bubble sort

$O(n!)$ = factorial time → extremely slow

- Travelling salesman problem

Bin code