Big O = Upper bound

$f(x) <= c.g(x)$

$\Rightarrow f(x) = O(g(x))$

For eg. $n - 1 <= c.n$

So, $n - 1 = O(n)$

Now, find time complexity of below given code :

```
flag = 0 ..... O(1)
for (int i=0; i<n; i++) ..... O(n)
{
if( A[i] == key )   .... O(n)
        flag=1  ... O(n) in worst case
}
```

O (1) + O(n) + O(n) + O(n) = O(n)

O(n) + O(n) +O(n)
= 2n + 3n + 4n (for example)
= 9 n
<= 10n, for all values of n
= O(n)
[Since, Big O is upper bound and we neglect constants in the analysis]

```
for (int i=0; i<n; i++)
{
    for(int j=0; j<i; j++)
    {
     .....
    }
}
```

i=0 => 0 time
i=1 => 1 time
i=2 => 2 time
.....

i=n-1 => n-1 times
Sum of total steps
= n * (n-1) /2
= (n^2 - n)/2
= O(n^2)
[Neglect lower powers]

```
for (int i=1; i<=n; i=i*2)
{
        ....
}
```

1, 2, 4, 8 .... <=n

no. of steps = log2 (n)

A quick note

include <bits/stdc++.h>

= includes all STL libraries in c++

Common Errors in online platforms

1. Compile time error (CE)

2. Wrong answer (WA)
eg. Yes ≠ YES .
So, read Input output format carefully

3. Time limit exceeded (TLE)
In general, Time Limit : 1 or 2 seconds
Improve the time complexity of code
Use fast I/O

4. Runtime error (RE) or
Segmentation Fault

FAST I/O with cin & cout

```
int main()
{
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);

.......
}
```

Reasons for Runtime Error (RE)

1. accessing out of bound index
of array .
eg. A[-1] , A[size], A[size+2]

2. divide by 0
eg. 4 / 0, a / 0

3. for overflow : (also gives WA sometimes)
int a=100000000, b=100000000;
int c= a*b; // 10^14 , which is >> 10^9 (the range of int in C++)

Preventing overflows

```
int a, b;
long long c = a + b ; // overflow still there

long long c = (long long)a + b ;
// now, okay

long long a,b;
c=a+b;
// this is also okay but better, as saves time
```

Another note

O (1) < O(log N) < O(root N)

< O(N) < O(N log N) < O(N^2) < O(N^3)

< O(N^4) ..... < O(2^N) < O (N^N)

In 1 second, 10^7 or 10^8 operations are performed :

Eg. O(N^2), N=10^3 // work
O (N log N) = N = 10^5 // work
O(N^2), N=10^5 // will give TLE

```
float a =3.99322;
float b=3.99322;
if( a==b)
cout<<"equal"; // bad practise
// For floating point comparisons,
// don't simply use ==
// You would get Wrong Answer
```

```
const float eps = 0.000001 ;//1e-6
float a,b;
if ( abs(a-b) <eps )
// good
// equality check
```

2.909000000001
2.909000000002