Time complexity

while loop

for loop

rabbit

every jump takes 1 sec

Total - 100 secs ⇒ N secs

$\frac{N}{2}$ secs

```
int a=5;
a = a+1;
int b=9;      b= a+2;
```

---

1) Take an input n.

2) Create an arr of size n.

Lets suppose, every line execution takes k unit of time.

total time = n*k order of n

O(n)

3) Increase value of every element of arr. by 1.

```
for(int i=0; i<n; i++){
    arr[i]++;
}
```

```
arr[0]++;
arr[1]++;
arr[2]++;
arr[3]++;
---- n
```

int a = 7; → k
int b = 7;
int n = → k

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

a = 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21
22 23 24 25 26

n = 5

$O(n^2)$

$n^2 k + k + k + k$

n^2

i = 0 , 1 , 2 , 3 , 4
↓ ↓ ↓ ↓ ↓
n n n n n

n * n → n^2

int n = 5; → k constant time

int a = 5 → k

for ( ) { } → $n^2 k$

for

$O(n^2)$

$n^2 k + k + k + k$ - -

$(n^2 + 1)\, k \Rightarrow n^2 k$

for (int i = 0; i < n; i = i + 2) {
  System.out.println(arr(i));
}

arr(0)
arr(2)
arr(4)
arr(6)

n

i = 0 2 4

$O\left(\frac{n}{2}\right) \Rightarrow O(n)$

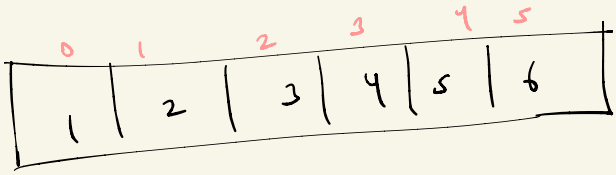
$O(2n) \Rightarrow O(n)$

$O(3n) \Rightarrow O(n)$

$O(kn) \Rightarrow O(n)$

k <<<< n

worst case scenario



tar = 11

```java
public static int search(int arr[], int n, int tar){
    for(int i=0; i<n; i++){
        if(arr[i]==tar){
            return i;
        }
    }

    // if my compiler at this line, i was not able to return any index
    return -1;
}
```

Space complexity

int a = 5;

int b = 7;

int n = scn.nextInt();

int[] arr = new int[2];

int[] arr = new int[n];

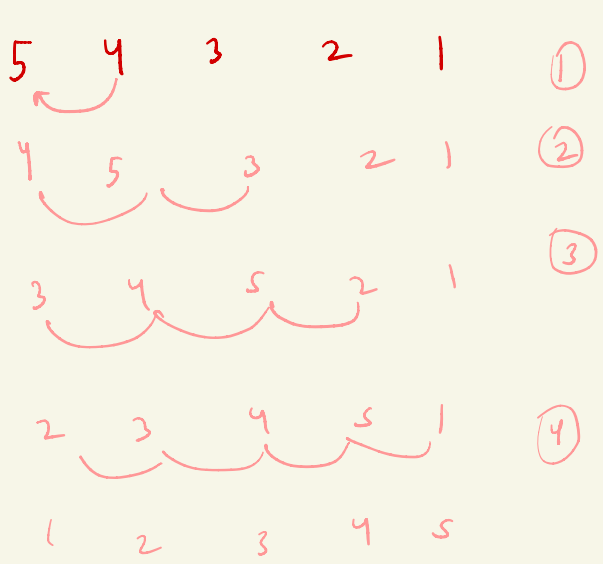
O(n) → space

---

1 second → 10^9 operations

$10^8 - 10^9$

$n \Rightarrow 10^3$



1, 2, 3, 4, 5, 6, 7

a=1 d=1

1+2+3+4+...5+... n

⇒ n(n+1)/2 ⇒ n²+n/2 ⇒ O(n²)

What is the time, space complexity of following code :

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    a = a + rand();
}
for (j = 0; j < M; j++) {
    b = b + rand();
}
```

O(N)

O(M)

time ⇒ O(N+M)

Assume that rand() is O(1) time, O(1) space function.

Constant space ⇒ O(1) space

pseudo code

What is the time, space complexity of following code :

constant space
↓
O(1)

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++) {
        a = a + j;
    }
}
for (k = 0; k < N; k++) {
    b = b + k;
}
```

$O(N^2)$

$O(N^2 + N)$

$O(N^2)$

$O(N)$

What is the time complexity of the following code :

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

$i = 100\ 50\ 25\ 12\ 6\ 3\ 1\ 0$

$a = 100\ 150\ 175\ 187\ 193\ 195\ 196$

number of terms $= x$

$N, \frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \frac{N}{16}, \ldots\ldots, 1$

$\frac{N}{2^0}, \frac{N}{2^1}, \frac{N}{2^2}, \frac{N}{2^3}, \frac{N}{2^4}, \ldots\ldots, \frac{N}{2^x}$

$\frac{N}{2^x} = 1 \Rightarrow N = 2^x$

$\Rightarrow \log_2 N = \log_2 2^x$

$\log_a a^b$

$\Rightarrow b \log_a a$

$\Rightarrow b$

$\Rightarrow \log_2 N = x \log_2 2$

$\Rightarrow \boxed{x = \log_2 N}$

N = 100

What is the time complexity of the following code :

```
int i, j, k = 0;
for (i = n/2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n/2;
    }
}
```

i = 50 ---------- 100

i = 50 → log n

i = 51 → log n

i = 52 → log n

i = 100 → log n

$\frac{N}{2} \log n$

$\rightarrow O(n \log n)$

number of terms = x

j = 2, 4, 8, 16, 32 ---- N

$2^1, 2^2, 2^3, 2^4, 2^5$ ---- 2^x

$2^x = N$

$\log_2 2^x = \log_2 N$

$x \log_2 2 = \log_2 N$

$\boxed{x = \log_2 N}$

$\log a^b$

$\Rightarrow b \log a$

$\log_a a = 1$

i += j → i = i + j

i = i + 2 → i += 2

a = a * 2 → a *= 2

C) for(i = 1; i < n; i *= 2) → log n

D) for(i = n; i > -1; i /= 2) infinite

n = 100

i = 1 2 4 8 16 32 64

i = 1 - - 4 8 16 - - - - n ⇒ log n

$2^0, 2^1, 2^2, - - - - 2^x$

$2^x = n \rightarrow x = \log n$

d) n = 100

i = 100 50 25 12 6 3 1 0 0 0 0 0 0 0 0

worst-case scenario

③ $f1 = 2^n$

② $f2 = \sqrt{n^3}$

① $f3 = n \log n$

④ $f4 = n^{\log n}$

$f(2) \Rightarrow \sqrt{n^3}$

$f(3) \Rightarrow n \log n$ ↓ larger value

(2), (log n)

$\Rightarrow n\sqrt{n}$

```
int a=1;
for (int i=1; i<=n; i++){
    for (int j=2; j*j<=n; j++){
        a++;
    }
}
```

$j^2 <= n$

$j <= \sqrt{n}$

$\sqrt{n}$

n = 100

j = 2 3 4 5 6 7 8 9 10 1

j = 9 10 $\sqrt{n}$

9 ⇒ 10 → 10-1

$2*2<100$

$3*3<100$

$4*4<100$

$5*5<100$

$6*6<100$

$7*7<100$

$8*8<100$

$9*9<100$

$10*10<100$