Time Complexities - Iterative

1.

```
int a = 0;
for (i = 0; i < N; i++) {
    a = a + rand(); // o(1) times
}
```

2.

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

3.

```
int a = 0;
    for (i = 0; i < N; i++) {
        for (j = N; j > i; j--) {
            a = a + i + j;
        }
    }
```

4.

```
// c can we anything other than 1
for (int i = 1; i <= n; i *= c){
    // some O(1) operations
}
for (int i = n; i > 0; i /= c){
    // some O(1) operations
}
```

5.

```
for(int i = 0; i*i < n; i++){
    pf("neeraj")
}
```

6.

```
i = 1, s = 1
while( s <= n){
    i = i + 1;
    s = s + i;
    print("neeraj")
}
```

7.

```
for(int i = 0; i < n; i++){
    for(int j = 0; j < i; j++){
        // constant time operations
  }
}
```

8.

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

9.

```
for(i = 1; i<= n; i++){
    for(int j = 1; j <= n; j++){
        if(j == 1)
            continue;
        print(“Neeraj”)
    }
}
```

10.

```
for(i = 1; i <= n; i++){
    for(j = 1; j <= n; j = j + i){
        print(“neeraj”);
    }
}
```

```
for(j = 1; j <= n; j = j + 3){
        print(“neeraj”);
    }
```

Time Complexities - Recursive

Fibonacci Series

```
fib(n)
{
    if(n == 0) then return 0
    else if(n == 1) then return 1
    else return fib(n-1) + fib(n-2)
}
```

Space Complexity

1.

```
A(n) {
  if(n >= 1){
    A(n-1);
    print("neeraj");
  }
}
```

2.

```
A(n) {
   if(n >= 1){
      A(n-1);
      print(“neeraj”);
      A(n-1);
   }
}
```

Find time and space

```
int f(int n) {
      if (n <= 1) {
            return 1;
      }
      return f(n - 1) + f(n - 1);
}
```