Time Complexity

What is efficiency in a program ?

In a program efficiency can be considered in 2 ways -

- Time Complexity :
  - There are trillions of data need to be searched and sort as per user search , so if time complexity will not considered then it might takes hours to fetch the results. `Google Search` : use to pagerank algorithm for searching millions of info in less than 1 sec.
- Space Complexity :
  - There are lot of space constrain in real world , so to maintain such constraint we should use space complexity concept. `Game Space` : Reducing a game storage from gb to mb for lite version of it so than it can be a part of user who have less storage phones.

In [3]:
```python
number = int(input('enter the number'))

digits = '0123456789'
result = ''
while number != 0:
  result = digits[number % 10] + result
  number = number//10

print(result)

# Log(n)
```

```
345465
```

In [4]:
```python
L = [1,2,3,4]

sum = 0
for i in L:
  sum = sum + i

product = 1
```

```
for i in L:
  product = product*i

print(sum,product)

# O(n)+O(n)=O(2n)==O(n)
```

```
10 24
```

In [6]:

```
A = [1,2,3,4]
B = [5,6,7,8]
for i in A:
  for j in B:
    print(i,j)

# O(len(A)*len(B))
```

```
1 5
1 6
1 7
1 8
2 5
2 6
2 7
2 8
3 5
3 6
3 7
3 8
4 5
4 6
4 7
4 8
```

In [8]:

```
A = [1,2,3,4]
B = [5,6,7,8]

for i in A:
  for j in B:
    for k in range(100):
      print(i,j)
```

```
# O(len(A)*len(B))
```

```
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
```

```
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
```

```
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 5
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
```

```
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
```

```
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 6
1 7
1 7
1 7
1 7
1 7
```

```
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
```

```
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
```

```
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 7
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
```

```
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
```

```
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
1 8
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
```

```
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
```

```
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
```

```
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 5
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
```

```
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
```

```
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 6
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
```

```
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
```

```
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
2 7
```

```
2 7
2 7
2 7
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
```

```
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
```

```
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
2 8
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
```

```
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
```

```
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 5
3 6
3 6
```

```
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
3  6
```

```
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
```

```
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 6
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
```

```
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
```

```
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 7
3 8
3 8
3 8
3 8
3 8
3 8
3 8
```

```
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
```

```
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
```

```
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
3 8
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
```

```
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
```

```
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 5
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
```

```
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
```

```
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
4 6
```

```
4 6
4 6
4 6
4 6
4 6
4 6
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
```

```
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
```

```
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 7
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
```

```
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
```

```
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
4 8
```

```
4 8
4 8
```

In [10]:
```
L = [1,2,3,4,5]

for i in range(0,len(L)//2):
  other = len(L) - i -1
  temp = L[i]
  L[i] = L[other]
  L[other] = temp

print(L)

# O(n/2)==O(n)
```

```
[5, 4, 3, 2, 1]
```

In [11]:
```
n = 10
k = 0
for i in range(n//2,n):
  for j in range(2,n,pow(2,j)):
        k = k + n / 2;

print(k)


# O(nlogn)
```

```
35.0
```

In [13]:
```
a = 10
b = 3

if b <= 0:
  print(-1)
div = a//b

print(a-div-b)

# O(1)
```

```
4
```

In [14]:
```python
n = 345

sum = 0
while n>0:
  sum = sum + n%10
  n = n // 10

print(sum)

# O(logn)
```

```
12
```

In [15]:
```python
def fib(n):
  if n == 1 or n == 0:
    return 1
  else:
    return fib(n-1) + fib(n-2)

# O(2^n)
```

In [ ]:
```python
# Subset Algo
```

```
           {3T(n-1) if n>0
    T(n) = {1, otherwise
```

In [16]:
```python
# O(3^n)
```

```
           {2T(n-1)-1 if n>0
    T(n) = {1, otherwise
```

In [17]:
```python
# O(1)
```

End