mad-pract-time-complexity

June 28, 2024

[1]:
```
# 1 CONSTANT TIME COMPLEXITY BY STEP COUNT
stepcount = 0
print("Hello World!!")
stepcount = stepcount+1
print("Time Complexity is of order = ",stepcount)
```

```
Hello World!!
Time Complexity is of order =  1
```

[2]:
```
# 2 LINEAR TIME COMPLEXITY BY STEP COUNT
stepcount = 0
n = 10
for i in range(1, n + 1):
  print("Hello World !!!")
  stepcount = i
print("Time Complexity is of order n = ",stepcount)
```

```
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Hello World !!!
Time Complexity is of order n =  10
```

[3]:
```
# 3 LOGARITHMIC COMPLEXITY
import time
start=time.time()
n = 8
# for (i = 1; i <= n; i=i*2) {
for i in range(1,9,+2):
  print("Hello World!!")
end=time.time()
print("Time Complexity is of the order log2(n) and time taken = ",(end-start))
```

```
Hello World!!
Hello World!!
Hello World!!
Hello World!!
Time Complexity is of the order log2(n) and time taken =  0.0002696514129638672
```

[4]:
```
# 4 LOGARITHMIC COMPLEXITY
start = time.time()
n = 50
i = 2
for j in range(2,n+1):
  if(i >= n):
    break
  print("Hello World !!!")
  i *= i
end = time.time()
print("Time Complexity is of the order log(log(n)) and time taken =␣
 ↪",(end-start))
```

```
Hello World !!!
Hello World !!!
Hello World !!!
Time Complexity is of the order log(log(n)) and time taken =
0.0072252750396728516
```

[5]:
```
# 5 Time Complexity for Sum Function Calling
import time
def sum_list(list): # 1 - t1
  start = time.time()
  total = 0 # 2 - t2
  for num in list: # length of the list say n each step takes time t
    total += num # 3 - t3
  end = time.time()
  return total ,1000*(end-start) # 4 - t4
list=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,17,18,19,20]
print("(sum, time taken(t1+t2+t*n+t3*n+t4)) =",sum_list(list))
```

```
(sum, time taken(t1+t2+t*n+t3*n+t4)) = (210, 0.002384185791015625)
```

[6]:
```
# 6 Time Complexity for Product Function Calling
import time
def product_list(list): # 1 - t1
  start = time.time()
  total = 1 # 2 - t2
  for num in list: # length of the list say n each step takes time t
    total *= num # 3 - t3
  end = time.time()
  return total ,1000*(end-start) # 4 - t4
```

```
list=[1, 2, 3, 4, 5]
print("(product, time taken(t1+t2+t*n+t3*n+t4)) =",product_list(list))
```

```
(product, time taken(t1+t2+t*n+t3*n+t4)) = (120, 0.0021457672119140625)
```

[7]:
```
# 7 Time Complexity for two for loops
import time
start = time.time()
my_list = [1, 2, 3, 4, 5] # t1
for i in range(len(my_list)): # t2
  for j in range(len(my_list)): # t3
    print(my_list[i], my_list[j]) # n * m * t4
end=time.time()
print("time taken = ",1000*(end-start))
```

```
1 1
1 2
1 3
1 4
1 5
2 1
2 2
2 3
2 4
2 5
3 1
3 2
3 3
3 4
3 5
4 1
4 2
4 3
4 4
4 5
5 1
5 2
5 3
5 4
5 5
time taken =  21.745681762695312
```

[8]:
```
# 8. Find the Sum of 2 numbers on the above machine
start=time.time()
def sum(a,b):
  return a+b
# function call
```

```
a = 10
b = 12
print(sum(a,b))
end= time.time()
print("time taken = ",1000*(end-start))
```

```
22
time taken =  1.5954971313476562
```

[ ]: