

Lab Assignment 1

Name: K sai Venkata Raju

BL.EN.U4CSE21089

Question 1) Find the sum of first N natural numbers using Iterative and Recursive algorithms. Find the time taken to execute the same by varying 'N's value and plot it using python's plot function.

Code)

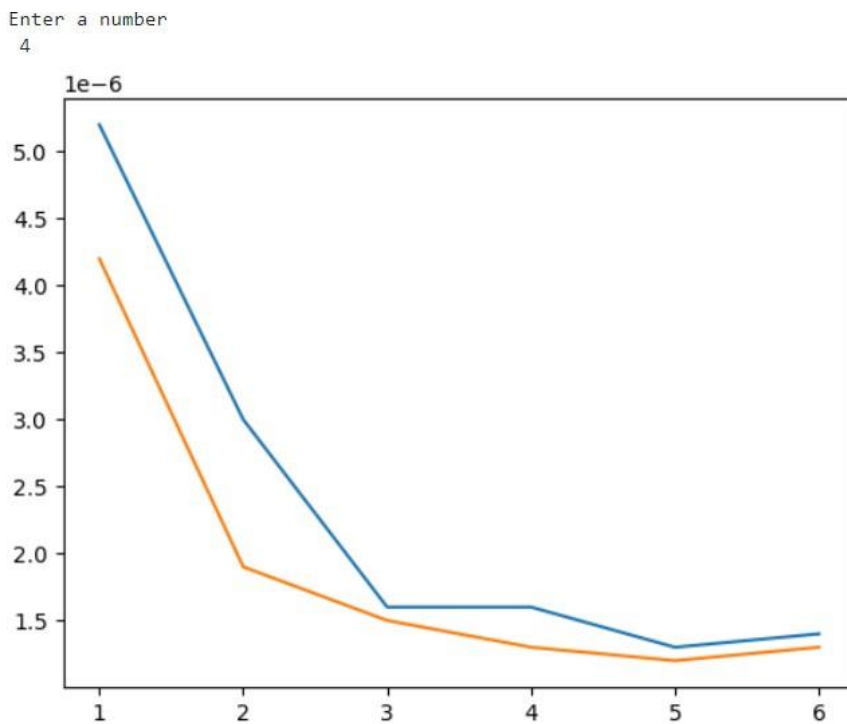
```
import time
import matplotlib.pyplot as plt
def sumofni(n):
    sum = 0
    for i in range(1,n+1):
        sum += i
    return sum
start =
time.perf_counter()
def sumofn(n):
    if n == 1:
        return
    1 else:
        return n + sumofn(n-
1)
ite_sum = []
rec_sum = []
n = int(input("Enter a number\n"))
for _ in range(6):
    st_time =
time.perf_counter()
    itsum =
sumofni(n)
    end_time = time.perf_counter()
    ite_sum.append(end_time - st_time)
```

```

st_time =
time.perf_counter() resum =
sumofni(n)
end_time = time.perf_counter()
rec_sum.append(end_time - st_time)
plt.plot(range(1,7), ite_sum, label = "Iterative
Sum") plt.plot(range(1,7), rec_sum, label =
"Recursive Sum")plt.show()

```

output)



Question 2) Perform linear and binary searches for an array of 10000 elements. Use random function in Python to generate the integer array elements in the range 1 to 1000. The search key is an input given by the user. Plot the time taken by the algorithm for 5 different searches when executing the two algorithms.

Code)

```

import time
import matplotlib.pyplot as plt

```

```
import random
```

```
def linsea(l, k):
```

```
    for i in l:
```

```
        if i ==
```

```
            k:
```

```
                return 1
```

```
    return 0
```

```
def binsea(l, k):
```

```
    left = 0
```

```
    right = len(l) - 1
```

```
    while left <= right:
```

```
        mid = (left + right) //
```

```
        2if l[mid] == k:
```

```
            return mid
```

```
        elif k < l[mid]:
```

```
            right = mid -
```

```
        1 else:
```

```
            left = mid +
```

```
    1 return -1
```

```
arr = [random.randint(1, 1000) for _ in
```

```
range(10000)]k = int(input("Enter search key: "))
```

```
linsearch = []
```

```
binsearch = []
```

```
for _ in range(5):
```

```
    st_time =
```

```
    time.perf_counter()li_re =
```

```
    linsea(arr, k)
```

```
    end_time = time.perf_counter()
```

```
linsearch.append(end_time - st_time)
```

```
arr.sort() # Sorting the array for binary search
```

```
s_time = time.perf_counter()
```

```
bi_re = binsea(arr, k)
```

```
en_time = time.perf_counter()
```

```
binsearch.append(en_time - s_time)
```

```
plt.plot(range(1, 6), linsearch, label="Linear
```

```
Search") plt.plot(range(1, 6), binsearch,
```

```
label="Binary Search")plt.xlabel("Search
```

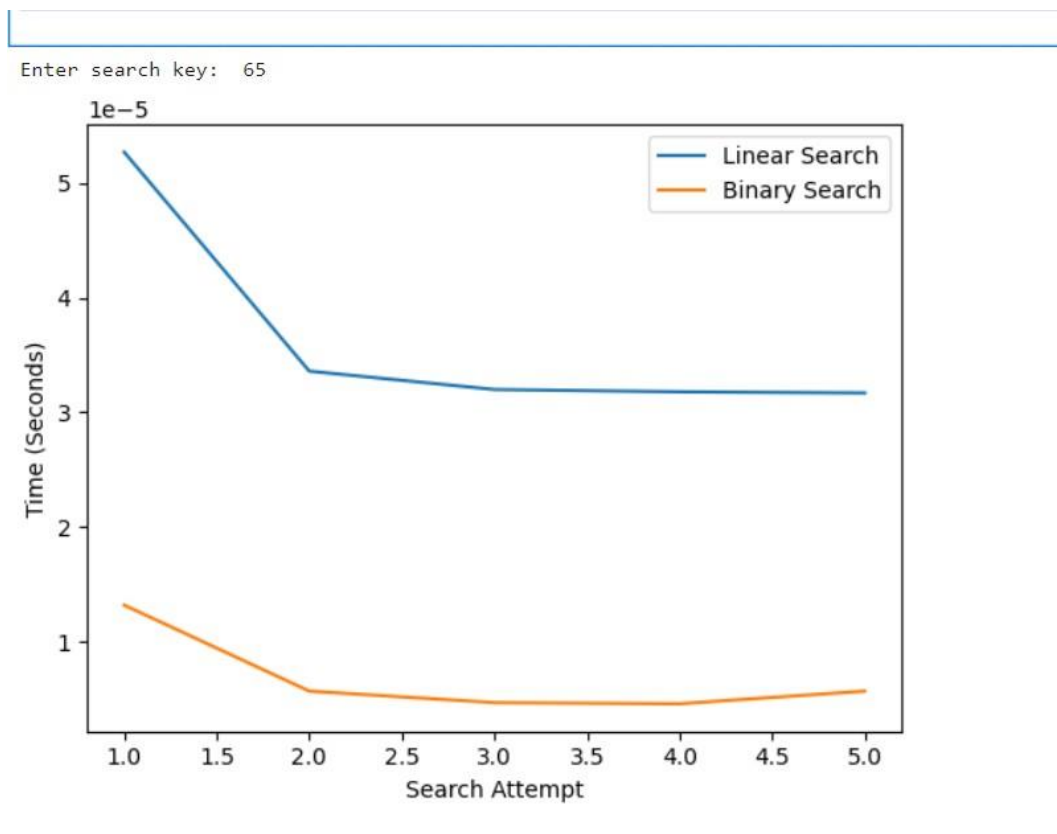
```
Attempt")
```

```
plt.ylabel("Time
```

```
(Seconds)")plt.legend()
```

```
plt.show()
```

Output)



Question 3) Write a recursive function to convert the entered string of digits into the integer it represents. For example, 13531 represents the integer 13,531.

Code)

```
def conver(st):  
    if len(st) <= 3:  
        return  
    stelse:  
        return conver(st[:-3]) + ',' + st[-3:]
```

```
n = input("Enter a  
string\n")print(n)  
s = conver(n)  
print(s)
```

Output)

```
Enter a string  
14589  
14589  
14,589
```

Question 4) Write a short recursive Python function that takes a character string s and outputs its reverse. For example, the reverse of pots&pans would be snap&stop.

Code)

```
def reverse_string(s):  
    # Base case: If the string is empty or has only one character, return  
    it as is if len(s) <= 1:  
        return s  
  
    # Recursive case: Swap the first and last characters and reverse the middle
```

```
return s[-1] + reverse_string(s[1:-1]) + s[0]
```

```
# Input a string
```

```
input_string = input("Enter a string: ")
```

```
reversed_string = reverse_string(input_string)
```

```
print(f"The reverse of '{input_string}' is: {reversed_string}")
```

Output)

```
Enter a string: sam&jam
The reverse of 'sam&jam' is: maj&mas
```

Question 5) Write a short recursive Python function that determines if a string s is a palindrome. For example, racecar and gohangasalamiimalasagnahog are palindromes

Code)

```
def is_palindrome(s):
```

```
# Base case: If the string has zero or one character, it's a
```

```
palindromeif len(s) <= 1:
```

```
    return True
```

```
# Check if the first and last characters are the same (case
```

```
insensitive)if s[0].lower() == s[-1].lower():
```

```
    # Recursively check the substring without the first and last
```

```
    charactersreturn is_palindrome(s[1:-1])
```

```
else:
```

```
    return False
```

```
# Input a string
```

```
input_string = input("Enter a string: ")
```

```
if is_palindrome(input_string):  
    print(f'{input_string}' is a  
palindrome.")else:  
    print(f'{input_string}' is not a palindrome.")
```

Output)

```
Enter a string:  aba  
'aba' is a palindrome.
```