NAME : SABARIVASAN V
REG NO : 205001085
CLASS : CSE - B

ASSIGNMENT 2

```
# Implement Factorial using recursion and draw a graph for the number
of steps
#  the algorithm takes to accomplish the task for different sizes of
n.
```

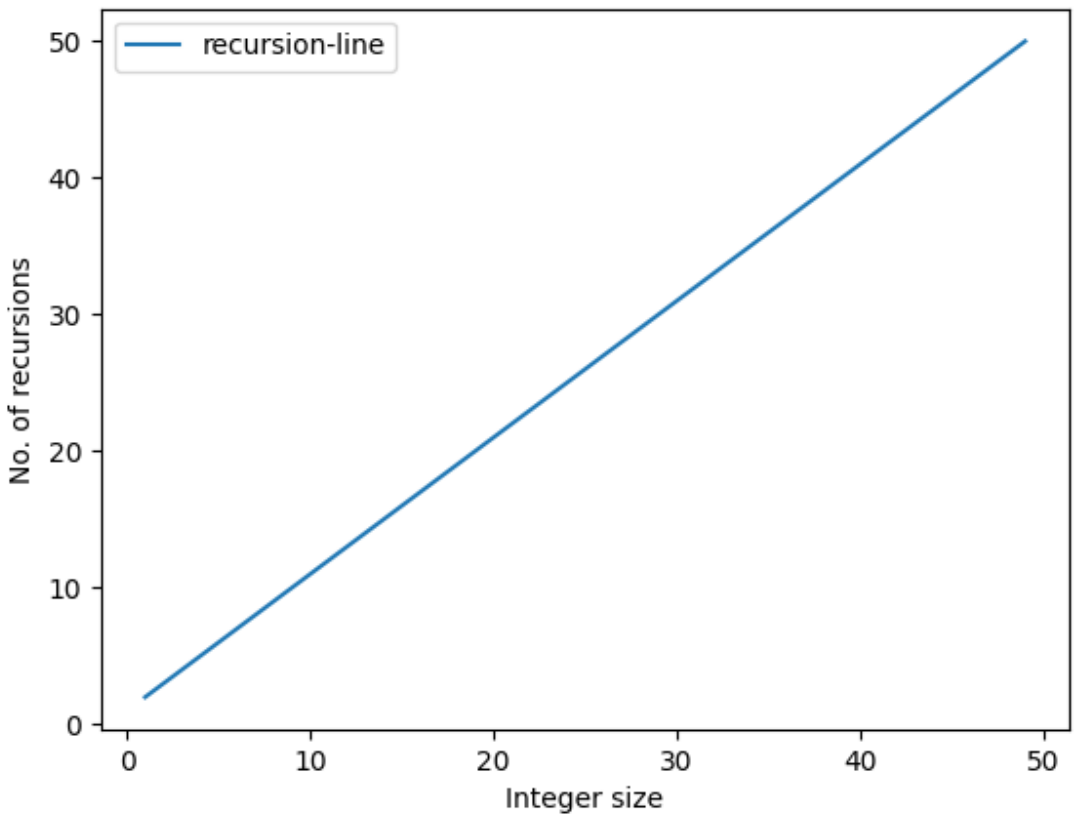
```
import matplotlib.pyplot as plt
```

```
def factorial(n,count):
    count+=1
    if n==0:
        return count
    return factorial(n-1,count)
```

```
xpoints = []
ypoints = []
```

```
for i in range(1,50):
    count = 0
    xpoints.append(i)
    ypoints.append(factorial(i,count))
```

```
plt.plot(xpoints, ypoints,label ='recursion-line')
plt.xlabel("Integer size")
plt.ylabel("No. of recursions")
plt.legend()
plt.show()
```

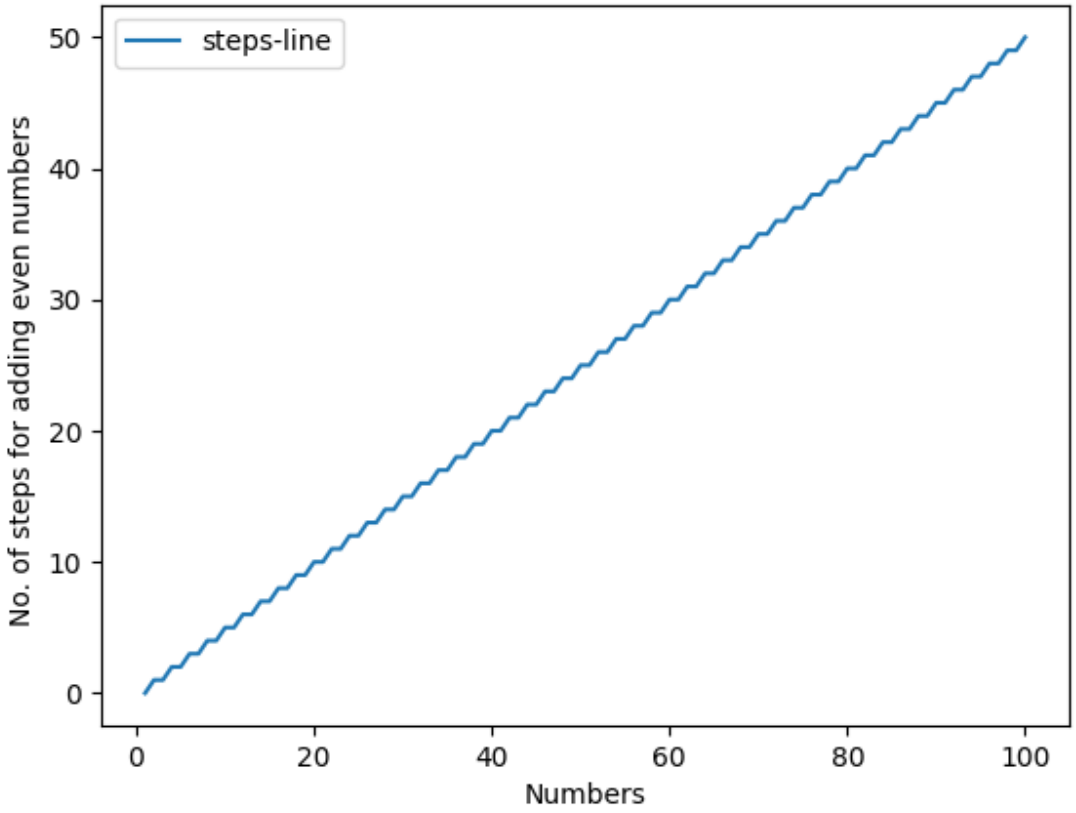

recursion-line
No. of recursions
Integer size
0
10
20
30
40
50

```
# Implement sum of even numbers and draw a graph for the number
# of steps the algorithm takes to accomplish the task for
different sizes of n.
# Compare the graph with function n
```

```
import matplotlib.pyplot as plt
```

```
xpoints = [i for i in range(1,101)]
ypoints = [j//2 for j in xpoints]
```

```
plt.plot(xpoints, ypoints,label ='steps-line')
plt.xlabel("Numbers")
plt.ylabel("No. of steps for adding even numbers ")
plt.legend()
plt.show()
```



```
# Implement a recursive function that generates a new
# list every time (size of the list=n) and performs
# binary search with a random number. Do this for
# n times. Draw a graph to illustrate the number of
# counts required to accomplish the task for different
#  sizes of n.

import numpy as np
import matplotlib.pyplot as plt
import random
from math import log

def bin_search(array,element):
   lower = 0
   higher = len(array) - 1
   m = count = 0

   while lower <= higher :
       count+=1
       m = (higher+lower)//2
       if array[m] < element :
           lower = m + 1
       elif array[m] > element :
           higher = m - 1
       else :
           return count
   return count

def func(n,array,count):
   if n==0:
       return count
   c = bin_search(array,random.randint(0, 1000))
```

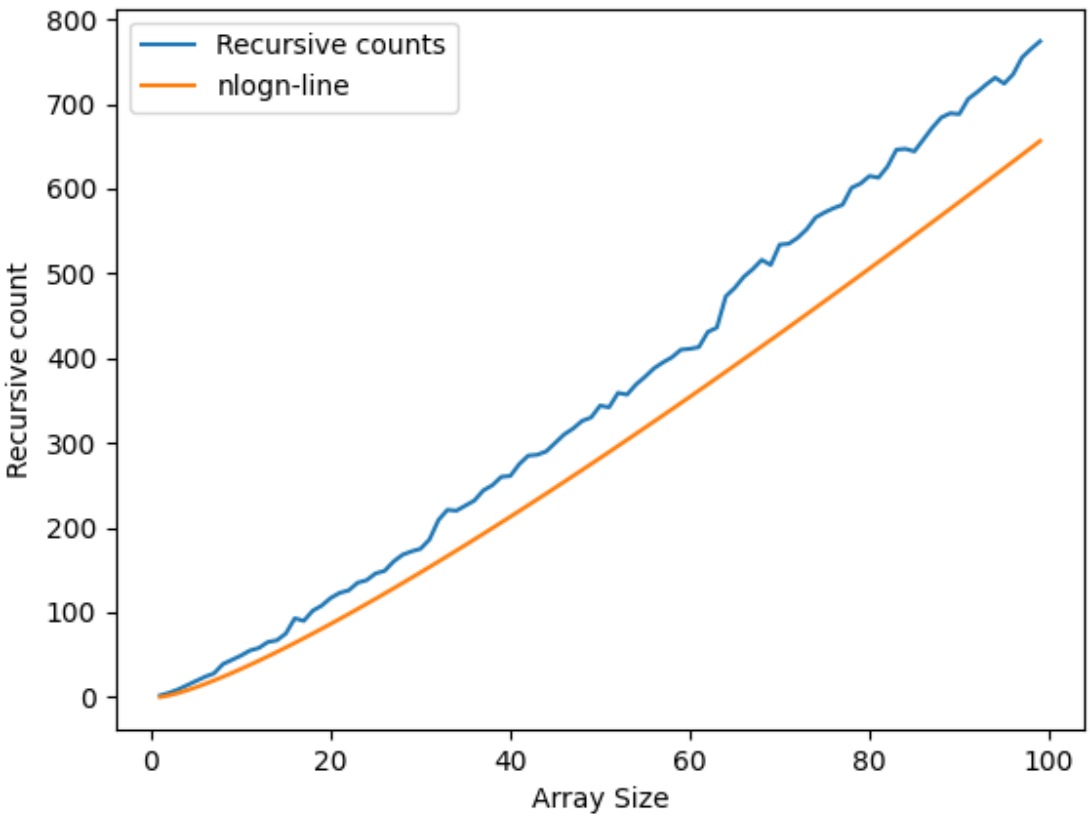
```
    return func(n-1,array,count+1+c)

xpoints = []
ypoints = []
# zpoints = []

for i in range(1,100):
    rand_int = np.random.randint(0,500,size=(i))
    rand_int.sort()
    xpoints.append(i)
    ypoints.append(func(i,rand_int,0))



plt.plot(xpoints, ypoints,label ='Recursive counts')
plt.plot(xpoints,[i*log(i,2) for i in range(1,100)],label
='nlogn-line')
plt.xlabel("Array Size")
plt.ylabel("Recursive count")
plt.legend()
plt.show()
```


Recursive counts
nlogn-line
Recursive count
Array Size
800
700
600
500
400
300
200
100
0
0
20
40
60
80
100