Lab2_problem_3

October 30, 2024

0.1 Problem 3

[1]:

```
##
import sys
import csv
import matplotlib.pyplot as plt
```

[7]:

```
#A
fibo = [1,1]
```

[8]:

```
#B
def fibo_append(fibo):
    next_fib = fibo[-1] + fibo[-2]
    fibo.append(next_fib)
```

[9]:

```
#C
fibo_append(fibo)
print (fibo)

# one more times just to check it that works properly and append another element

fibo_append(fibo)
print (fibo)
```

```
[1, 1, 2]
[1, 1, 2, 3]
```

[6]:

```
#D

for _ in range (1000000 - len(fibo)):
    fibo_append(fibo)
```

[7]:

```
#Verifying occupies significant memory
fibo_total_size = sum(sys.getsizeof(num) for num in fibo)
print("Total memory used in bytes:", fibo_total_size)
```

```
Total memory used in bytes: 46308685756
```

[8]:

```
## second part of d

#increase the limit for maximum digit in integer string conversion
sys.set_int_max_str_digits(10**7)

def fibonacci_large(n):
    a, b = 0,1
    for _ in range(n):
        a,b = b , a+b
    return a

Fp7= fibonacci_large (10**7)

#count digit
num_digits = len(str(Fp7))
print("Number of digits in F(10^7):", num_digits)
```

```
Number of digits in F(10^7): 2089877
```

[10]:

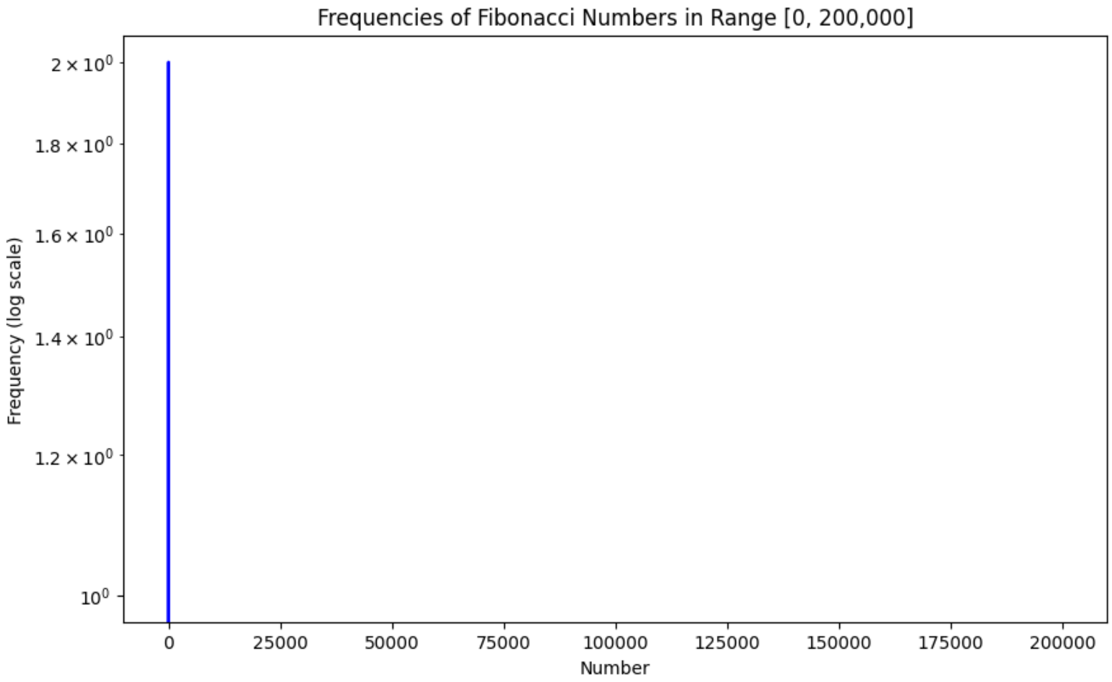
```
# Save F (10^7) to a text file
with open("Fibonacci_10^7.txt", "w") as file:
    file.write(str(Fp7))
```

[10]:

```
## E

counts = {i: 0 for i in range(200001)}
for number in fibo:
    if number <= 200000:
        counts[number] += 1


with open("fibonacci_counts.csv", "w", newline="") as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(["Number", "Count"])
    for number, count in counts.items():
        writer.writerow([number, count])
```

[12]:

```
numbers = list(counts.keys())
frequencies = list(counts.values())

plt.figure(figsize=(10, 6))
plt.semilogy(numbers, frequencies, 'b-')
plt.xlabel("Number")
plt.ylabel("Frequency (log scale)")
plt.title("Frequencies of Fibonacci Numbers in Range [0, 200,000]")
plt.show()

smallest_missing = next(i for i in range(200001) if counts[i] == 0)
```


Frequencies of Fibonacci Numbers in Range [0, 200,000]
Frequency (log scale)
2×10^0
1.8×10^0
1.6×10^0
1.4×10^0
1.2×10^0
10^0
0
25000
50000
75000
100000
125000
150000
175000
200000
Number