Program

With multithreading

import threading

import queue

import time

import sys

sys.setrecursionlimit(10000)

print("With Threading")

start\_time = time.time()

def fib(x, stop):

if(x < stop):

return 0

#print(f"{threading.currentThread().getName()} value {x} \n")

return x+fib(x-1, stop)

que = queue.Queue()

threads\_list = list()

x = 2100

h1 = x

h2 = int(abs(x/2))

stop = h2+1

fib\_thread1 = threading.Thread(target=lambda q, arg1: q.put(fib(arg1, stop)), args=(que, h1))

fib\_thread1.start()

threads\_list.append(fib\_thread1)

fib\_thread2 = threading.Thread(target=lambda q, arg1: q.put(fib(arg1, 0)), args=(que, h2))

fib\_thread2.start()

threads\_list.append(fib\_thread2)

for t in threads\_list:

t.join()

total = 0

while not que.empty():

result = que.get()

total = total + result

print(total)

print("Time taken %s seconds" % (time.time() - start\_time))

Without Multithreading

import time

import sys

sys.setrecursionlimit(10000)

print("Without Threading")

start\_time = time.time()

def fib(x):

if(x < 1):

return 0

return x+fib(x-1)

print(fib(2100))

print("Time taken %s seconds" % (time.time() - start\_time))

