CSE 5520 Fall 2021

Homework 1 (Due Tue 9/7/2021 midnight at HuskyCT)

Topics: Bar graph, dynamic plot and web publishing of plots

You have seen how to plot a bar graph for Fibonacci sequence in which the sequence is fixed. Your exercise is to write an extension of that code which enables you to interactively produce the bar graph based on entry of the number of terms entered. Sample Fibonacci number generation code is available in
https://www.programiz.com/python-programming/examples/fibonacci-sequence
It has been modified to a function as shown below.

```
def fibonacci(Nterms: int, N1: int, N2: int):
    Count = 0
    # check if the number of terms is valid
    if Nterms <= 0:
       print("Please enter a positive integer")
    # if there is only one term, return n1
    elif Nterms == 1:
       print("Fibonacci sequence upto",Nterms,":")
       print(N1)
    # generate fibonacci sequence
    else:
       print("Fibonacci sequence:")
       while Count < Nterms:
           print(N1)
           Nth = N1 + N2
           # update values
           N1 = N2
           N2 = Nth
           Count += 1
    return N1

nterms = int(input("How many terms? "))
fibonacci(nterms, 0, 1
```

Step 1: Produce two different graphs, one for 10 and 50 using pygal for the plots

Some sample code generating bar graph using pygal is available in
http://www.pygal.org/en/latest/index.html
.
Step 2: Repeat Step 1, this time using plotly.

Some sample code generating bar graph using plotly is available in
https://plotly.com/python/getting-started/

Step 3: Publish your plotly code in Step 2 using Dash in your local web browser http://127.0.0.1:8050/.

Some sample code using Dash with plotly is available in
https://dash.plotly.com/layout

You will need to kill Dash server if needed and the code is given below.

```
# https://stackoverflow.com/questions/15562446/how-to-stop-
flask-application-without-using-ctrl-c
# kill_server.py; kill Dash server running.

import os
import subprocess
import re

port = 8050
host = '127.0.0.1'
cmd_newlines = r'\r\n'

host_port = host + ':' + str(port)
pid_regex = re.compile(r'[0-9]+$')

netstat = subprocess.run(['netstat', '-n', '-a', '-o'],
stdout=subprocess.PIPE)
# Doesn't return correct PID info without precisely these flags
netstat = str(netstat)
lines = netstat.split(cmd_newlines)

for line in lines:
    if host_port in line:
        pid = pid_regex.findall(line)
        if pid:
            pid = pid[0]
            os.system('taskkill /F /PID ' + str(pid))

# And finally delete the .pyc cache
os.system('del /S *.pyc')
```

You upload only one PDF format including your code and output plots demonstrating your code works. Your report name should be "Lastname.HW1".

The report can include figures, tables and graphs with type-written in 8 x11 inch paper with 1 inch margin top, bottom and side. There is no page limit in your report.

HWs and Projects, 5% penalty for one day late submission. No acceptance after 5 days late. Extension is allowed only with the supporting medical record.