page rank algorithm :

```
import numpy as n
from fractions import Fraction as f

dp = f(1,3)
m = n.matrix([[0,0,1],[f(1,2),0,0],[f(1,2),1,0]])
ex = n.zeros((3,3))
ex[:] = dp
b = 0.7
a= b * m + ((1-b) * ex)
r = n.matrix([dp, dp, dp]).transpose()
prev_r = r
for i in range(1,100):
    r = a*r
    res=n.round((r).astype(float), decimals=3)
    print (res)
    if (prev_r==r).all():
        break
    prev_r = r
print ("Final:", res)
print ("sum", n.sum(r))
```

Aim: Implement Dynamic programming algorithm for computing the edit distance between strings s1
and s2. (Hint. Levenshtein Distance)

```
def dist(X, x, Y, y):
    if x == 0:
        return y
    if y == 0:
        return x
    c = 0 if (X[x - 1] == Y[y - 1]) else 1
    return min(dist(X, x - 1, Y, y) + 1, dist(X, x, Y, y - 1) + 1, dist(X, x - 1, Y, y - 1) + c)

X = 'kitten'
Y = 'sitting'
print('The Levenshtein distance is', dist(X, len(X), Y, len(Y)))
```

Write a program to Compute Similarity between two text documents

```
import math as m
import string as s

def freq(file_path):
    with open(file_path, 'r') as file:
        content = file.read()
    words = content.translate(str.maketrans(s.punctuation + s.ascii_uppercase, " " * len(s.punctuation) + s.ascii_lowercase)).split()
    word_freq = {}
    for word in words:
        word_freq[word] = word_freq.get(word, 0) + 1
    print("File", file_path, ":", len(content), "lines,", len(words), "words,", len(word_freq), "distinct words")
    return word_freq

def dot_product(dict1, dict2):
    dp = 0.0
    for word, freq in dict1.items():
```

```
        if word in dict2:
            dp += freq * dict2[word]
    return dp

document1 = freq('hello.txt')
document2 = freq('world.txt')

numerator = dot_product(document1, document2)
denominator = m.sqrt(dot_product(document1, document1) * dot_product(document2,
document2))

distance_radians = m.acos(numerator / denominator)
print("distance between documents is", distance_radians, "radians")
```

Aim: Program to count Uppercase, lowercase & Special characters

```
t = "HelloWorld12345@%%%%"
u, l, d, s = 0, 0, 0, 0
for char in t:
    if char.isupper():
        u += 1
    elif char.islower():
        l += 1
    elif char.isdigit():
        d += 1
    else:
        s += 1
print('upper case letters:', u, '\nlower case letters:', l, '\ndigits:', d, '\
nspecial characters:', s)
```

Write a program to implement simple web crawler
Steps:
1. Open cmd
Type the following commands
- python -m pip install requests
- python -m pip install bs4
- python -m pip install lxml

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urlparse

s = BeautifulSoup(requests.get("https://www.amazon.in").text, features="lxml")
for link in s.find_all("a"):
    href = link.get("href")
    if href:
        parsed_url = urlparse(href)
        if parsed_url.scheme and parsed_url.netloc:
            print(href)
```

Write a program for pre-processing of a text document : stop word removal

Open cmd
Type the following commands
- pip install regex

- pip install --user -U nltk

2. open vs code
3. create 2 files by clicking on the icon shown in the picture :
   - test.txt
   - pr_7stopwords.py

4. type the following in test.txt:

```
hello this is 1234
MY NAME IS 12333ABCC?//"'`"
but this kite rest the most and the beautiful of the world
```

```
import re
import string
import nltk

nltk.download('punkt')
nltk.download('stopwords')

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

file_content = open('hello.txt', 'r').read()
tokens = word_tokenize(file_content)

def remove_numbers(text):
    return re.sub(r'[0-9]', '', text)

print("Enter a choice to process your text: \n1: Lowercase \n2: Uppercase \n3:
Tokenize \n4: Remove numbers \n5: Remove punctuation \n6: Remove spaces \n7:
Remove stopwords")

while True:
    choice = int(input("Enter your choice (1-7): "))
    if choice == 1:
        print(file_content.lower())
    elif choice == 2:
        print(file_content.upper())
    elif choice == 3:
        print(tokens)
    elif choice == 4:
        print(remove_numbers(file_content))
    elif choice == 5:
        print(file_content.translate(str.maketrans('', '', string.punctuation)))
    elif choice == 6:
        print(file_content.replace(" ", ""))
    elif choice == 7:
        stop_words = set(stopwords.words('english'))
        print([token for token in tokens if token.lower() not in stop_words])
    else:
        print("Please enter a choice between 1-7")
```

Aim: Write a program to parse xml text, generate web graph and compute topic specific page rank

Steps:

1. Open cmd

Type the following commands

- pip install networkx

2. open VS code and type the following

```
import networkx as net
from lxml import etree as e

xml = '''
<root>
  <node id="1">cats</node>
  <node id="2">dogs</node>
  <node id="3">cats and dogs</node>
  <link from="1" to="3" />
  <link from="2" to="3" />
  <link from="3" to="1" />
</root>
'''

g = net.DiGraph()
for l in e.fromstring(xml).findall(".//link"):
    g.add_edge(l.attrib["from"], l.attrib["to"])

pr = net.pagerank(g, weight="weight", alpha=0.85).items()

for i in pr:
    if "cats" in i:
        pr[i] += 0.15

for node, score in pr:
    print(f"Node {node}: PageRank = {score}")
```