SIES College of Arts, Science & Commerce

RISE WITH EDUCATION

NAAC REACCREDITED - ‘A’ GRADE

ISO 9001 : 2008

S.I.E.S College of Arts, Science and Commerce(Autonomous)
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that Miss./Mr _**Ajay Kumar Uthaya Kumar**___Roll No. **TCS2324002** has successfully completed the necessary course of experiments in the subject of **Information Retrieval** during the academic year **2023 – 2024** complying with the requirements of **University of Mumbai**, for the course of **TYBSc Computer Science [Semester-VI].**

Prof. In-Charge
Rajesh Yadav

Examination date:

Examiner’s Signature & Date:

Head of the Department
Dr. Manoj Singh

College Seal

Index Page

Sr. No	Description	Page No	Date	Faculty Signature
1	Write a python program to demonstrate bitwise operation	3	23/12/2023	
2	Implementation of PageRank	8	03/01/2024	
3	Write a program to implement Levenshtein Distance.	11	09/01/2024	
4	Write a program to compute Similarity between two documents	13	09/01/2024	
5	Write a python program for map reducer.	16	24/01/2024	
6	Write a python program for HITS Algorithm	19	07/02/2024	
7	Write a python program for StopWords	21	24/01/2024	
8	Write a program for mining twitter to identify tweets for a specific period and identify trends and named entities.	24	24/01/2024	
9	Write a python program to implement simple web crawling.	27	29/01/2024	
10	Write a program to implement XML Retrieval.	30	07/02/2024	
11	Write a program to retrieve xml test using xml library.	33	07/02/2024	
12	Write a program to retrieve xml test using lxml library.	35	07/02/2024	

Name of Instructor: Mr.Rajesh Yadav

SIES

Information Retrieval

Practical No.1

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Bitwise Operator	**Batch**	I
Date :	21/12/23	Practical No	1

A) AIM:

Write a python program to demonstrate bitwise operation.

Bitwise operations in Python are commonly used for low-level programming and manipulation of individual bits.

B) DESCRIPTION:

1. The & (bitwise AND) in C or C++ takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1.
2. The | (bitwise OR) in C or C++ takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 if any of the two bits is 1.
3. The ^ (bitwise XOR) in C or C++ takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different.
4. The << (left shift) in C or C++ takes two numbers, the left shifts the bits of the first operand, and the second operand decides the number of places to shift.
5. The >> (right shift) in C or C++ takes two numbers, right shifts the bits of the first operand, and the second operand decides the number of places to shift.
6. The ~ (bitwise NOT) in C or C++ takes one number and inverts all bits of it.

The term-document incidence matrix is one of the basic techniques to represent text data where, we get the unique words across all the documents. For each document, we add 1 if the term exists in the document otherwise 0 in the cell

C) CODE AND OUTPUT:

1st Method:

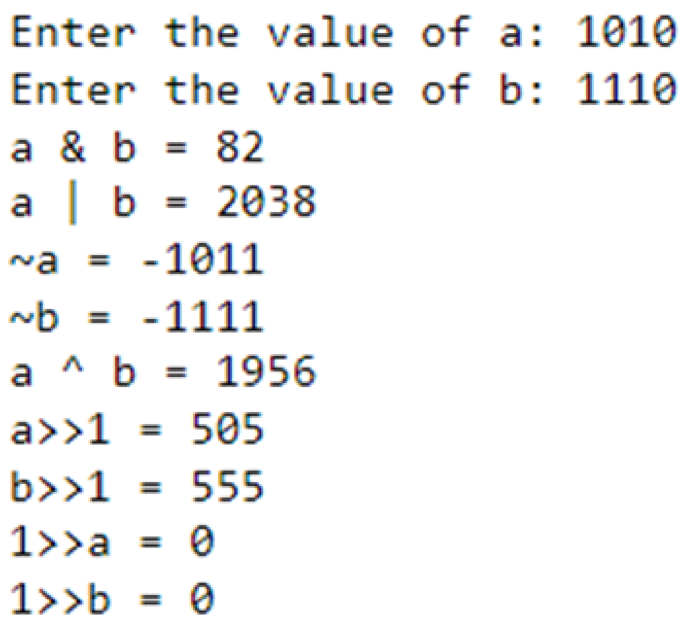
```
def bitwise_operations(a,b):
    bitwise_and_result = a & b
    print("a & b =", bitwise_and_result)

    bitwise_or_result = a | b
    print("a | b =", bitwise_or_result)

    bitwise_not_resulta =  ~a
    print("~a =", bitwise_not_resulta)

    bitwise_not_resultb = ~b
    print("~b =", bitwise_not_resultb)

    bitwise_xor_result = a ^ b
    print("a ^ b =", bitwise_xor_result)

    bitwise_rightshift_resulta = a >> 1
    print("a >> 1 =", bitwise_rightshift_resulta)

    bitwise_rightshift_resultb = b >> 1
    print("b >> 1 =", bitwise_rightshift_resultb)


    bitwise_leftshift_resulta = 1 >> a

    print("1 >> a =", bitwise_leftshift_resulta)


    bitwise_leftshift_resultb = 1 >> b

    print("1 >> b =", bitwise_leftshift_resultb)


a = int(input("Enter the value of a: "))

b = int(input("Enter the value of b: "))


bitwise_operations(a,b)
```

2nd Method

```
import pandas as pd

from sklearn.feature_extraction.text import CountVectorizer

print('Boolean RetrievalModal Using Bitwise Operations on Term Document Incidence
Matrix\n')

corpus={'this is the first document','this is the second document','and this is the third
document','Is this the first document'}

print("The corupus is: \n",corpus)

vectorizer = CountVectorizer()

x = vectorizer.fit_transform(corpus)

df = pd.DataFrame(x.toarray(),columns=vectorizer.get_feature_names_out())

print("\nThe generated data frame\n")

print(df)

print("\nQuery processing on Term Document Incidence Matrix")
```

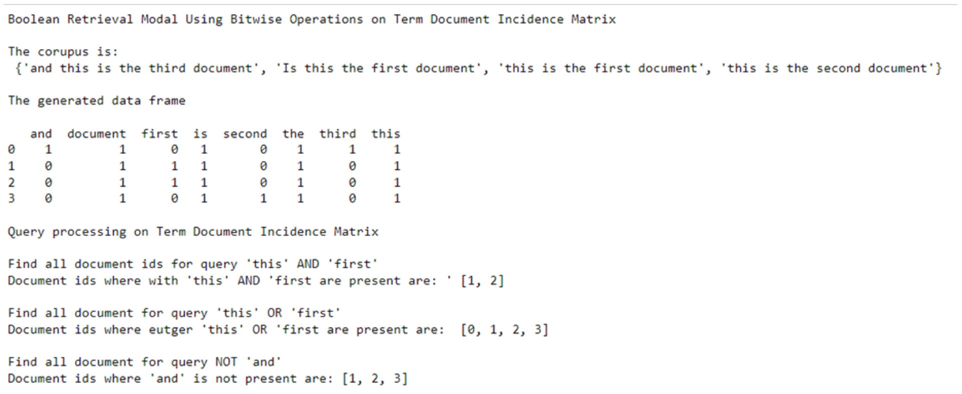
```
#AND

print("\nFind all document ids for query 'this' AND 'first'")

alldata = df[(df['this']==1)&(df['first']==1)]

print("Document ids where with 'this' AND 'first are present are: '", alldata.index.tolist())


#OR

print("\nFind all document for query 'this' OR 'first'")

alldata = df[(df['this']==1)|(df['first']==1)]

print("Document ids where eutger 'this' OR 'first are present are: ", alldata.index.tolist())


#NOT

print("\nFind all document for query NOT 'and'")

alldata = df[(df['and']!=1)]

print("Document ids where 'and' is not present are:", alldata.index.tolist())
```

1st Method:



```
Enter the value of a: 1010
Enter the value of b: 1110
a & b = 82
a | b = 2038
~a = -1011
~b = -1111
a ^ b = 1956
a>>1 = 505
b>>1 = 555
1>>a = 0
1>>b = 0
```

2nd Method:



```
Boolean Retrieval Modal Using Bitwise Operations on Term Document Incidence Matrix

The corupus is:
 {'and this is the third document', 'Is this the first document', 'this is the first document', 'this is the second document'}

The generated data frame

   and  document  first  is  second  the  third  this
0    1         1      0   1       0    1      1     1
1    0         1      1   1       0    1      0     1
2    0         1      1   1       0    1      0     1
3    0         1      0   1       1    1      0     1

Query processing on Term Document Incidence Matrix

Find all document ids for query 'this' AND 'first'
Document ids where with 'this' AND 'first are present are: ' [1, 2]

Find all document for query 'this' OR 'first'
Document ids where eutger 'this' OR 'first are present are:  [0, 1, 2, 3]

Find all document for query NOT 'and'
Document ids where 'and' is not present are: [1, 2, 3]
```

Information Retrieval
Practical No.2

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	PageRank	**Batch**	I
Date :	3/1/24	Practical No	2

A) AIM:
Implement page rank

B) DESCRIPTION:
NetworkX: NetworkX is a Python package or the creation, manipulation of the structure, dynamics, and functions of complex networks.
PyLab: PyLab is a convenience module that bulk imports matplotlib.pyplot (for plotting) and NumPy (for Mathematics and working with arrays) in a single name space. Although many examples use PyLab it is no longer recommended. Installation of the PyLab Module is installed at as the matplotlib package.
By the networkx package in python we can calculate page rank like below.

C) CODE AND OUTPUT:
1st Method – Without using NetworkX

```
def page_rank(graph, damping_factor=0.85, max_iterations=100, tolerance=1e-6):
    num_pages = len(graph)
    initial_page_rank = 1.0 / num_pages
    page_ranks = {page:initial_page_rank for page in graph}
    for _ in range(max_iterations):
        new_page_ranks = {}
        for page in graph:
            new_rank = (1-damping_factor)/num_pages
            for link in graph:
                if page in graph[link]:
                    new_rank += damping_factor * (page_ranks[link]/len(graph[link]))
            new_page_ranks[page] = new_rank
        convergence = all(abs(new_page_ranks[page] - page_ranks[page]) <  tolerance for page
in graph)
        if convergence:
            break
        page_ranks = new_page_ranks
    return page_ranks

if __name__ == "__main__":
```

```
graph = {
    'A':['B','C'],
    'B':['A'],
    'C':['A','B'],
    'D':['B']
}
result=page_rank(graph)
for page, rank in sorted(result.items(),key=lambda x: x[1], reverse=True):
    print(f"Page: {page} - PageRank: {rank:.4f}")
```

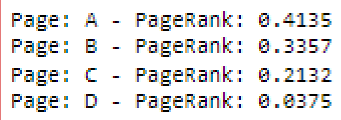
2nd Method – Using NetworkX with weighted graph

```
import networkx as nx
import pylab as plt
G = nx.DiGraph()
G.add_weighted_edges_from([('A','B',1),('A','C',1),('C','A',1),('B','C',1)])
ppr1 = nx.pagerank(G)
print("Page rank value: ", ppr1)
pos = nx.spiral_layout(G)
nx.draw(G, pos, with_labels=True, node_color="#f86e00")
plt.show()
```

3rd Method – Using NetworkX without weighted graph

```
import networkx as nx
#import matplotlib.pyplot as plt
import pylab as plt
G = nx.DiGraph()
[G.add_node(k) for k in ["A", "B", "C", "D", "E", "F", "G"]]
G.add_edges_from([('G', 'A'), ('A', 'G'), ('B', 'A'),
                  ('A', 'C'), ('C', 'A'), ('F', 'A'),
                  ('E', 'A'), ('A', 'D'), ('D', 'F'),
                 ('D','B')])
ppr1 = nx.pagerank(G)
print("Page rank value: ", ppr1)
pos = nx.spiral_layout(G)
nx.draw(G, pos, with_labels=True, node_color="#f86e00")
plt.show()
```

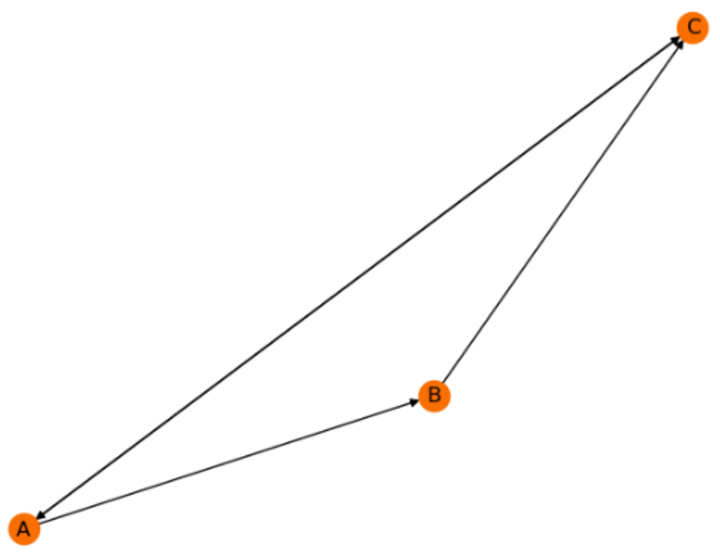
1st Method – Without Using NetworkX



```
Page: A - PageRank: 0.4135
Page: B - PageRank: 0.3357
Page: C - PageRank: 0.2132
Page: D - PageRank: 0.0375
```

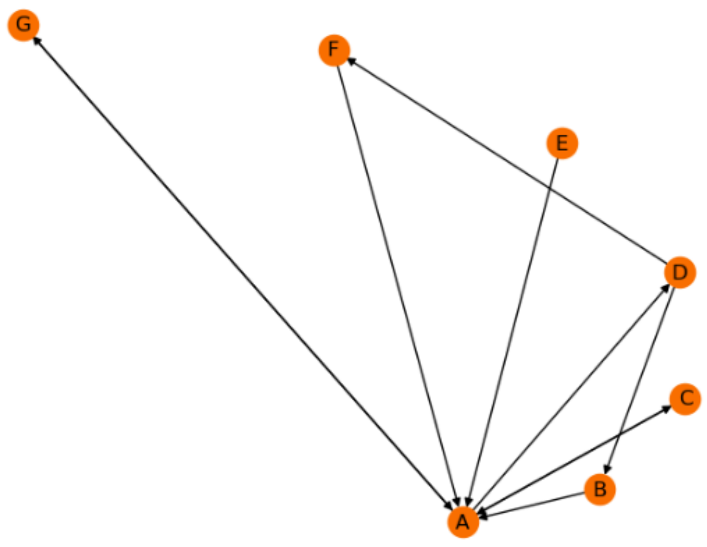
2nd Method – Using NetworkX with weighted Graph

```
Page rank value:  {'A': 0.387789442707259, 'B': 0.21481051315058508, 'C': 0.3974000441421556}
```



3rd Method – Using NetworkX without weighted Graph

```
Page rank value:  {'A': 0.4080745143467559, 'B': 0.07967426232810562, 'C': 0.13704946318948705, 'D': 0.13704946318948705, '
0.021428571428571432, 'F': 0.07967426232810562, 'G': 0.13704946318948705}
```



Information Retrieval

Practical No.3

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Levenshtein Distance	**Batch**	I
Date :	9/1/24	Practical No	1

A) AIM:

Write a program to implement Levenshtein Distance.

B) DESCRIPTION:

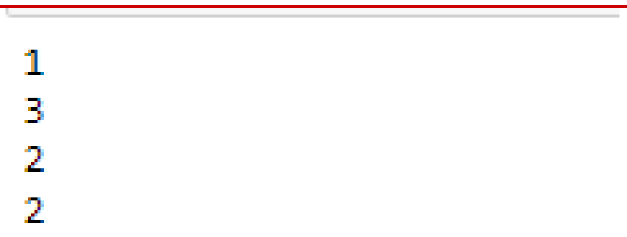
Levenshtein distance is a measure of the similarity between two strings, which takes into account the number of insertion, deletion and substitution operations needed to transform one string into the other.

Operations in Levenshtein distance are:

- Insertion: Adding a character to string A.
- Deletion: Removing a character from string A.
- Replacement: Replacing a character in string A with another character.

C) CODE AND OUTPUT:

```
def leven(x, y):

    n = len(x)

    m = len(y)
    A = [[i+j for j in range(m + 1)]for i in range(n + 1)]

    for i in range(n):

        for j in range(m):

            A[i+1][j+1] = min(A[i][j + 1]+1,

                              A[i + 1][j]+1,

                              A[i][j]+int(x[i]!=y[j]))

    return A[n][m]

print(leven("brap","rap"))

print(leven("trial","try"))

print(leven("horse","force"))

print(leven("rose","erode"))
```



```
1
3
2
2
```

Information Retrieval

Practical No.4

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Jaccard and Cosine Similarity	**Batch**	I
Date :	9/1/24	Practical No	1

A) AIM:

Write a program to compute Similarity between two documents
1) Jaccard Similarity
2) Cosine similarity

B) DESCRIPTION:

Jaccard Similarity: Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. Jaccard similarity can be used to find the similarity between two asymmetric binary vectors or to find the similarity between two sets. In literature, Jaccard similarity, symbolized by, can also be referred to as Jaccard Index, Jaccard Coefficient, Jaccard Dissimilarity, and Jaccard Distance.

Cosine Similarity: Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

C) CODE AND OUTPUT:

1. Jaccard Similarity

```
def Jaccard_Similarity(doc1, doc2):

    words_doc1 = set(doc1.lower().split())

    words_doc2 = set(doc2.lower().split())
```

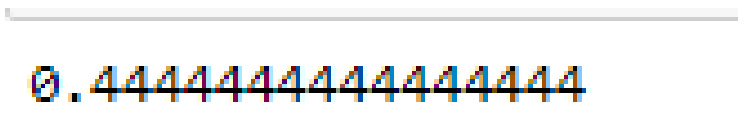
```
0.4444444444444444
```

```
    intersection = words_doc1.intersection(words_doc2)

    union = words_doc1.union(words_doc2)

    return float(len(intersection))/len(union)

doc_1 = "Data is the new oil of the digital economy"

doc_2 = "Data is a new oil"

Jaccard_Similarity(doc_1, doc_2)
```

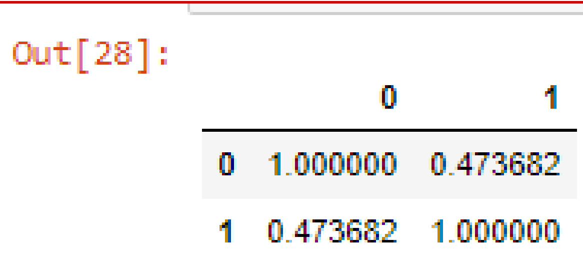
2. Cosine Similarity

```
import pandas as pd

import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine_similarity


doc_1 = "Data is the new oil of the digital economy"

doc_2 = "Data is a new oil"

data = [doc_1,doc_2]

Tfidf_vect = TfidfVectorizer()

vector_matrix = Tfidf_vect.fit_transform(data)

tokens = Tfidf_vect.get_feature_names_out()

pd.DataFrame(vector_matrix.toarray(),columns=tokens)

cosine_similarity_matrix  = cosine_similarity(vector_matrix)

pd.DataFrame(cosine_similarity_matrix)
```

1. Jaccard Similarity


```
0.4444444444444444
```

2. Cosine Similarity



Out[28]:

	0	1
0	1.000000	0.473682
1	0.473682	1.000000

3.

SIES

Information Retrieval

Practical No.5

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	MapReducer	**Batch**	I
Date :	24/1/24	Practical No	5

A) AIM:

Write a python program to implement map-reducer program to count the number of occurrence of each alphabetic character in a given dataset. The count for each letters should be case-insensitive

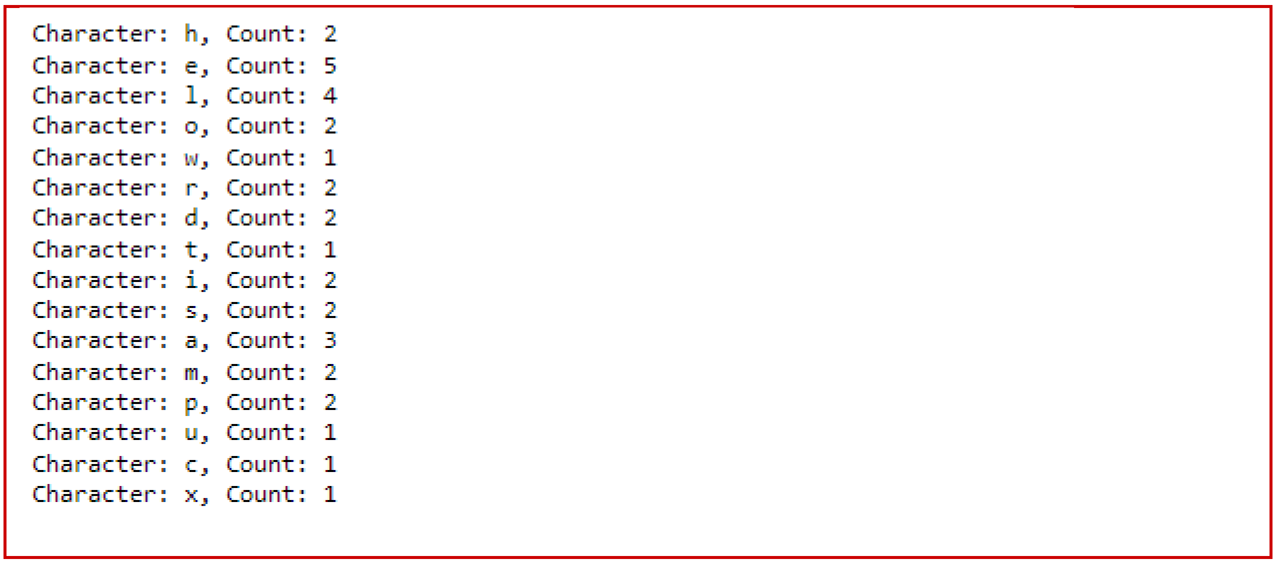
B) DESCRIPTION:

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job. The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

Name of Instructor: Mr.Rajesh Yadav

C) CODE AND OUTPUT:

```
from functools import reduce

from collections import defaultdict

def mapper(data):

    char_count = defaultdict(int)

    for char in data:

        if char.isalpha():

            char_count[char.lower()] += 1

    return char_count.items()

def reducer(counts1, counts2):

    merged_counts = defaultdict(int)

    for char, count in counts1:

        merged_counts[char] += count

    for char, count in counts2:

        merged_counts[char] += count

    return merged_counts.items()


if __name__ =="__main__":

    dataset = "Hello World! This is a MapReduce example."

    chunks = [chunk for chunk in dataset.split()]

    mapped_results = map(mapper, chunks)

    final_counts = reduce(reducer, mapped_results)

    for char, count in final_counts:

        print(f"Character: {char}, Count: {count}")
```



```
Character: h, Count: 2
Character: e, Count: 5
Character: l, Count: 4
Character: o, Count: 2
Character: w, Count: 1
Character: r, Count: 2
Character: d, Count: 2
Character: t, Count: 1
Character: i, Count: 2
Character: s, Count: 2
Character: a, Count: 3
Character: m, Count: 2
Character: p, Count: 2
Character: u, Count: 1
Character: c, Count: 1
Character: x, Count: 1
```

Name of Instructor: Mr.Rajesh Yadav

Information Retrieval

Practical No.6

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Hits Algorithm	**Batch**	I
Date :	7/2/24	Practical No	6

A) AIM:

Write a python program to implement HITS Algorithm.

B) DESCRIPTION:

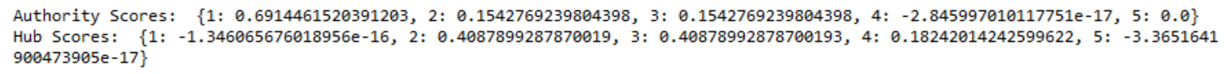
Hyperlink Induced Topic Search (HITS) Algorithm is a Link Analysis Algorithm that rates webpages, developed by Jon Kleinberg. This algorithm is used to the web link-structures to discover and rank the webpages relevant for a particular search. HITS uses hubs and authorities to define a recursive relationship between webpages. Before understanding the HITS Algorithm, we first need to know about Hubs and Authorities.

- Given a query to a Search Engine, the set of highly relevant web pages are called Roots. They are potential Authorities.
- Pages that are not very relevant but point to pages in the Root are called Hubs. Thus, an Authority is a page that many hubs link to whereas a Hub is a page that links to many authorities

C) CODE AND OUTPUT:

```
import networkx as nx

G = nx.DiGraph()
```

```
G.add_edges_from([(1,2),(1,3),(2,4),(3,4),(4,5)])

authority_scores, hub_scores = nx.hits(G)

print("Authority Scores: ",authority_scores)

print("Hub Scores: ",hub_scores)
```



```
Authority Scores:  {1: 0.6914461520391203, 2: 0.1542769239804398, 3: 0.1542769239804398, 4: -2.845997010117751e-17, 5: 0.0}
Hub Scores:  {1: -1.346065676018956e-16, 2: 0.4087899287870019, 3: 0.40878992878700193, 4: 0.18242014242599622, 5: -3.3651641
900473905e-17}
```

Name of Instructor: Mr.Rajesh Yadav

SIES

Information Retrieval

Practical No.7

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Stopwords	**Batch**	I
Date :	24/1/24	Practical No	7

A) AIM:

Write a python program to pre-processing of a text document: stop words removal.

B) DESCRIPTION:

A stop word is a commonly used word (such as “the”, “a”, “an”, or “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

This tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. It must be trained on a large collection of plaintext in the target language before it can be used. The NLTK data package includes a pre-trained Punkt tokenizer for English.

Name of Instructor: Mr.Rajesh Yadav

C) CODE AND OUTPUT:

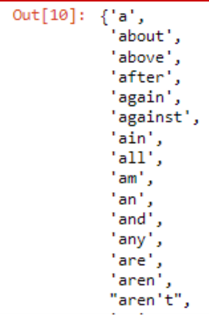
1) To download Stopwords

```
import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

set(stopwords.words('english'))
```

2) To Tokenize and Filter out Stopwords

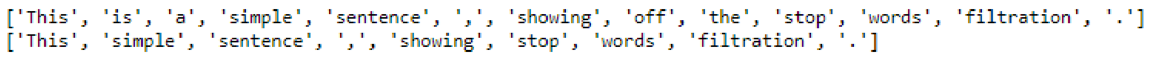
```
from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

example_sent = "This is a simple sentence, showing off the stop words filtration."

stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example_sent)

filtered_sentence = [w for w in word_tokens if not w in stop_words]

filtered_sentence = []

for w in word_tokens:

    if w not in stop_words:

        filtered_sentence.append(w)

print(word_tokens)

print(filtered_sentence)
```

1) To download Stopwords



```
Out[10]: {'a',
          'about',
          'above',
          'after',
          'again',
          'against',
          'ain',
          'all',
          'am',
          'an',
          'and',
          'any',
          'are',
          'aren',
          "aren't",
```

2) To Tokenize and Filter out Stopwords



```
['This', 'is', 'a', 'simple', 'sentence', ',', 'showing', 'off', 'the', 'stop', 'words', 'filtration', '.']
['This', 'simple', 'sentence', ',', 'showing', 'stop', 'words', 'filtration', '.']
```

SIES

Information Retrieval

Practical No.8

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Twitter Scrapping	**Batch**	I
Date :	24/1/24	Practical No	8

A) AIM:

Write a python program for mining Twitter to identify tweets for a specific period and identify trends and named entities.

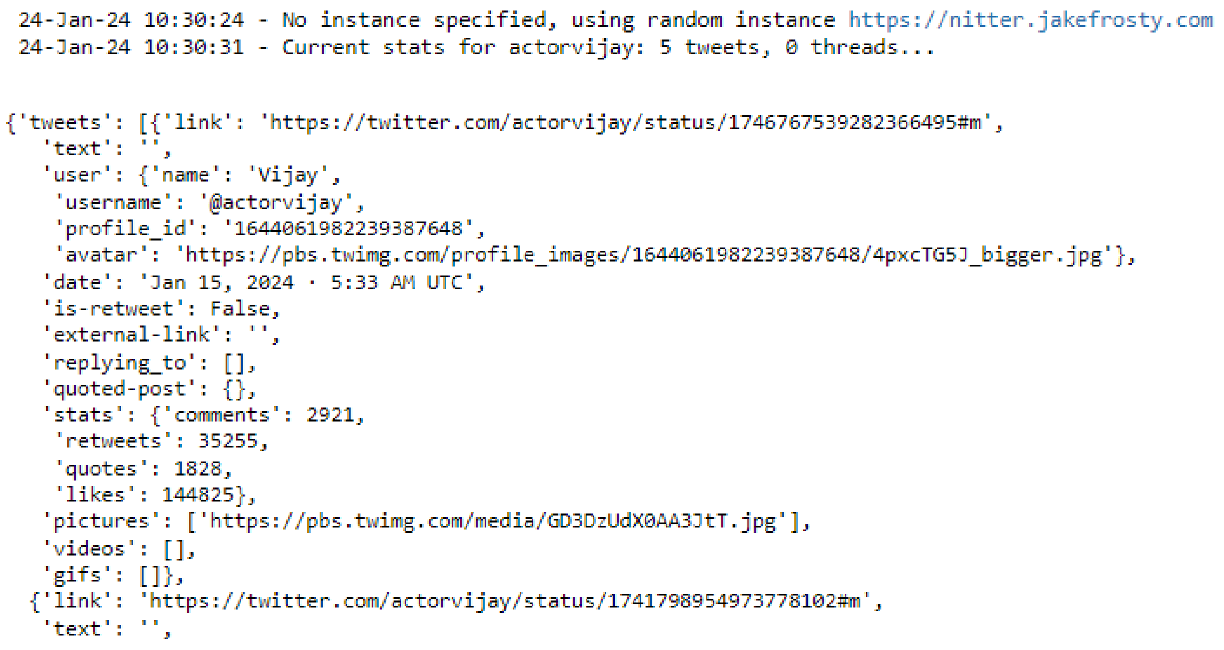
B) DESCRIPTION:

Nitter is a free and open source alternative viewer for Twitter/X, focusing on privacy and performance. Its minimalist and unaugmented UI resembles the classic Twitter desktop layout. Since the user cannot log in to Twitter through Nitter, Nitter has no notifications, no home feed and no ability to tweet. By default Nitter has no infinite scroll hence doomscrolling is unlikely. In addition to the official web instance, there are unofficial public web instances, as well as community-contributed mobile apps and browser extensions (for which there is also a list on Nitter's wiki) so browsers can auto-redirect Twitter URLs, and even randomize viewing requests across multiple instances. Nitter is funded by donations as well as a grant from NLnet's NGI fund

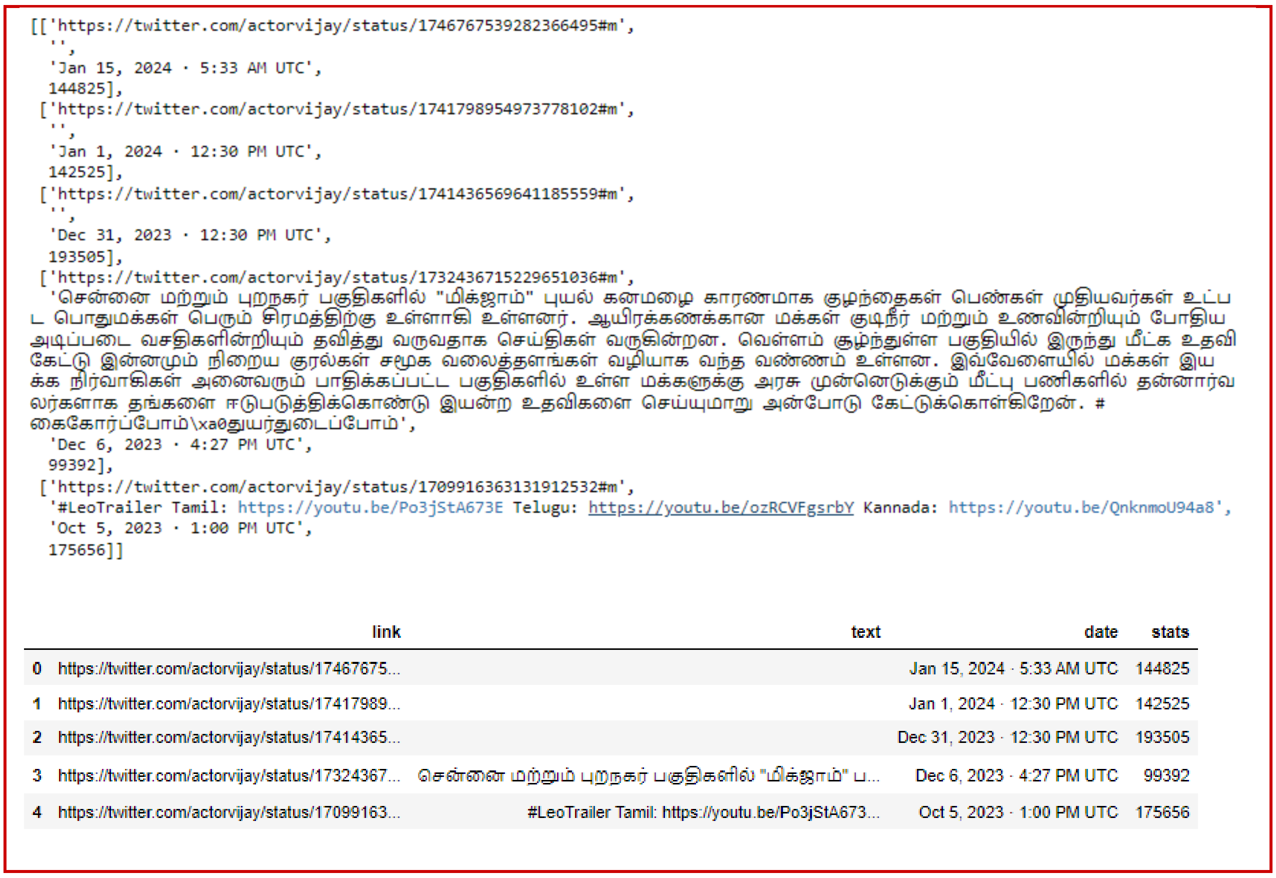
C) CODE AND OUTPUT:

```
import pandas as pd

from ntscraper import Nitter
```

```
scraper = Nitter()

tweets = scraper.get_tweets('actorvijay', mode='user', number=5)

tweets

final_tweets=[]

for tweet in tweets['tweets']:

   data = [tweet['link'],tweet['text'],tweet['date'],tweet['stats']['likes']]

   final_tweets.append(data)

final_tweets

data = pd.DataFrame(final_tweets,

                    columns=['link','text','date','stats'])

data
```



```
 24-Jan-24 10:30:24 - No instance specified, using random instance https://nitter.jakefrosty.com
 24-Jan-24 10:30:31 - Current stats for actorvijay: 5 tweets, 0 threads...


{'tweets': [{'link': 'https://twitter.com/actorvijay/status/1746767539282366495#m',
   'text': '',
   'user': {'name': 'Vijay',
    'username': '@actorvijay',
    'profile_id': '1644061982239387648',
    'avatar': 'https://pbs.twimg.com/profile_images/1644061982239387648/4pxcTG5J_bigger.jpg'},
   'date': 'Jan 15, 2024 · 5:33 AM UTC',
   'is-retweet': False,
   'external-link': '',
   'replying_to': [],
   'quoted-post': {},
   'stats': {'comments': 2921,
    'retweets': 35255,
    'quotes': 1828,
    'likes': 144825},
   'pictures': ['https://pbs.twimg.com/media/GD3DzUdX0AA3JtT.jpg'],
   'videos': [],
   'gifs': []},
  {'link': 'https://twitter.com/actorvijay/status/1741798954973778102#m',
   'text': '',
```



```
[['https://twitter.com/actorvijay/status/1746767539282366495#m',
  '',
  'Jan 15, 2024 · 5:33 AM UTC',
  144825],
 ['https://twitter.com/actorvijay/status/1741798954973778102#m',
  '',
  'Jan 1, 2024 · 12:30 PM UTC',
  142525],
 ['https://twitter.com/actorvijay/status/1741436569641185559#m',
  '',
  'Dec 31, 2023 · 12:30 PM UTC',
  193505],
 ['https://twitter.com/actorvijay/status/1732436715229651036#m',
  'சென்னை மற்றும் புறநகர் பகுதிகளில் "மிக்ஜாம்" புயல் கனமழை காரணமாக குழந்தைகள் பெண்கள் முதியவர்கள் உட்ப
ட பொதுமக்கள் பெரும் சிரமத்திற்கு உள்ளாகி உள்ளனர். ஆயிரக்கணக்கான மக்கள் குடிநீர் மற்றும் உணவின்றியும் போதிய
அடிப்படை வசதிகளின்றியும் தவித்து வருவதாக செய்திகள் வருகின்றன. வெள்ளம் சூழ்ந்துள்ள பகுதியில் இருந்து மீட்க உதவி
கேட்டு இன்னமும் நிறைய குரல்கள் சமூக வலைத்தளங்கள் வழியாக வந்த வண்ணம் உள்ளன. இவ்வேளையில் மக்கள் இய
க்க நிர்வாகிகள் அனைவரும் பாதிக்கப்பட்ட பகுதிகளில் உள்ள மக்களுக்கு அரசு முன்னெடுக்கும் மீட்பு பணிகளில் தன்னார்வ
லர்களாக தங்களை ஈடுபடுத்திக்கொண்டு இயன்ற உதவிகளை செய்யுமாறு அன்போடு கேட்டுக்கொள்கிறேன். #
கைகோர்ப்போம்\xa0துயர்துடைப்போம்',
  'Dec 6, 2023 · 4:27 PM UTC',
  99392],
 ['https://twitter.com/actorvijay/status/1709916363131912532#m',
  '#LeoTrailer Tamil: https://youtu.be/Po3jStA673E Telugu: https://youtu.be/ozRCVFgsrbY Kannada: https://youtu.be/QnknmoU94a8',
  'Oct 5, 2023 · 1:00 PM UTC',
  175656]]
```

	link	text	date	stats
0	https://twitter.com/actorvijay/status/17467675...		Jan 15, 2024 · 5:33 AM UTC	144825
1	https://twitter.com/actorvijay/status/17417989...		Jan 1, 2024 · 12:30 PM UTC	142525
2	https://twitter.com/actorvijay/status/17414365...		Dec 31, 2023 · 12:30 PM UTC	193505
3	https://twitter.com/actorvijay/status/17324367...	சென்னை மற்றும் புறநகர் பகுதிகளில் "மிக்ஜாம்" ப...	Dec 6, 2023 · 4:27 PM UTC	99392
4	https://twitter.com/actorvijay/status/17099163...	#LeoTrailer Tamil: https://youtu.be/Po3jStA673...	Oct 5, 2023 · 1:00 PM UTC	175656

SIES

Information Retrieval

Practical No.9

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Web Crawler	**Batch**	I
Date :	29/1/24	Practical No	9

A) AIM:

Write a python program to implement a simple web crawler

B) DESCRIPTION:

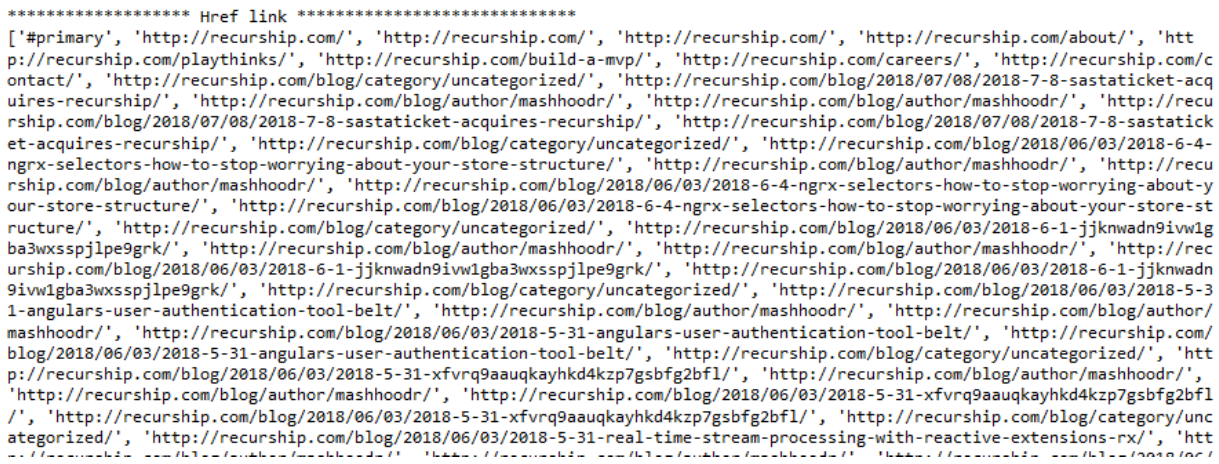
A web crawler is a digital search engine bot that uses copy and metadata to discover and index site pages. Also referred to as a spider bot, it "crawls" the world wide web to learn what a given page is about. It then indexes the pages and stores the information for future searches.

Working of web crawler:

1. The crawler begins with one or more URLs that constitute a seed set.
2. It picks a URL from this seed set, and then fetches the web pages at that URL.
3. The fetched page is then parsed, to extract both the text and the links from the page.
4. The extracted text is fed to a text indexer.
5. The extracted links and then added to a URL frontier, which at all times consists of URLs whose corresponding pages have yet to be fetched by the crawler.
6. Initially, the URL frontier contains the seed set; as pages are fetched, the corresponding URLs are deleted from the URL frontier. The entire process may be viewed as traversing the web graph,.

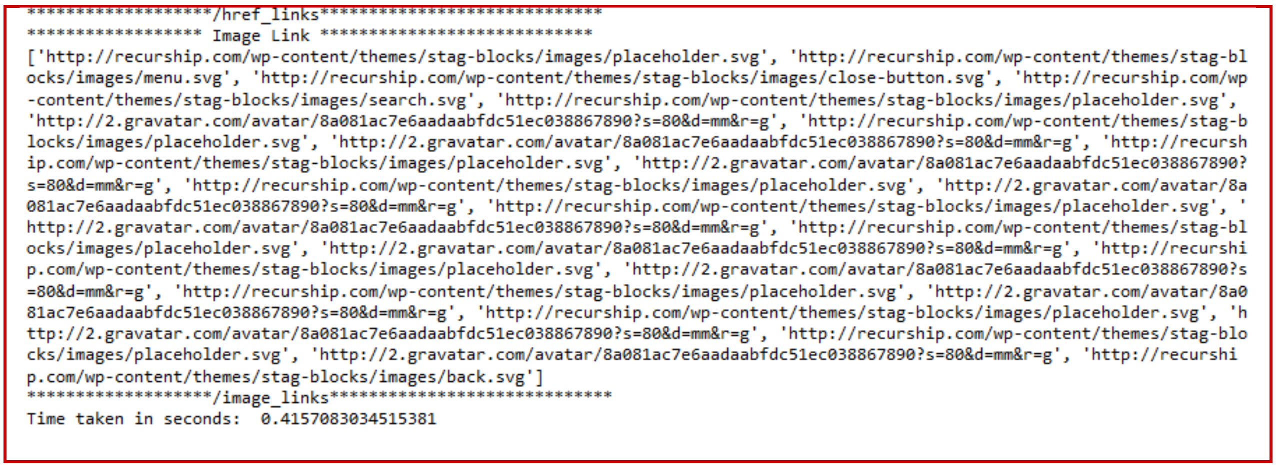
C) CODE AND OUTPUT:

```
import requests

from parsel import Selector

import time

start = time.time()

response = requests.get('http://recurship.com/')

selector = Selector(response.text)

href_links = selector.xpath('//a/@href').getall()

image_links = selector.xpath('//img/@src').getall()

print("****************** Href link *****************************")

print(href_links)

print("******************/href_links*****************************")

print("****************** Image Link ****************************")

print(image_links)

print("******************/image_links*****************************")

end=time.time()

print("Time taken in seconds: ",(end-start))
```



```
****************** Href link *****************************
['#primary', 'http://recurship.com/', 'http://recurship.com/', 'http://recurship.com/', 'http://recurship.com/about/', 'htt
p://recurship.com/playthinks/', 'http://recurship.com/build-a-mvp/', 'http://recurship.com/careers/', 'http://recurship.com/c
ontact/', 'http://recurship.com/blog/category/uncategorized/', 'http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acq
uires-recurship/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recu
rship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/', 'http://recurship.com/blog/2018/07/08/2018-7-8-sastatick
et-acquires-recurship/', 'http://recurship.com/blog/category/uncategorized/', 'http://recurship.com/blog/2018/06/03/2018-6-4-
ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recu
rship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-y
our-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-st
ructure/', 'http://recurship.com/blog/category/uncategorized/', 'http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1g
ba3wxsspjlpe9grk/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/author/mashhoodr/', 'http://rec
urship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1gba3wxsspjlpe9grk/', 'http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn
9ivw1gba3wxsspjlpe9grk/', 'http://recurship.com/blog/category/uncategorized/', 'http://recurship.com/blog/2018/06/03/2018-5-3
1-angulars-user-authentication-tool-belt/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/author/
mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/', 'http://recurship.com/
blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/', 'http://recurship.com/blog/category/uncategorized/', 'htt
p://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauqkayhkd4kzp7gsbfg2bfl/', 'http://recurship.com/blog/author/mashhoodr/',
'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauqkayhkd4kzp7gsbfg2bfl
/', 'http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauqkayhkd4kzp7gsbfg2bfl/', 'http://recurship.com/blog/category/unc
ategorized/', 'http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/', 'htt
```

Name of Instructor: Mr.Rajesh Yadav



```
*******************/href_links*****************************
****************** Image Link ****************************
['http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'http://recurship.com/wp-content/themes/stag-bl
ocks/images/menu.svg', 'http://recurship.com/wp-content/themes/stag-blocks/images/close-button.svg', 'http://recurship.com/wp
-content/themes/stag-blocks/images/search.svg', 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg',
'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-b
locks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recursh
ip.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?
s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a
081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', '
http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-bl
ocks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurshi
p.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s
=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a0
81ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'h
ttp://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blo
cks/images/placeholder.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'http://recurshi
p.com/wp-content/themes/stag-blocks/images/back.svg']
*******************/image_links*****************************
Time taken in seconds:  0.4157083034515381
```

SIES

Information Retrieval

Practical No.10

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	XML Retrieval	**Batch**	I
Date :	7/2/24	Practical No	10

A) AIM:

Write a python program to parse XML text, generate Web graph and compute topic specific page rank.

B) DESCRIPTION:

An Extensible Markup Language (XML) file is a text-based document that you can save with the .xml extension. You can write XML similar to other text files. To create or edit an XML file, you can use any of the following: Text editors like Notepad or Notepad++ Online XML editors.

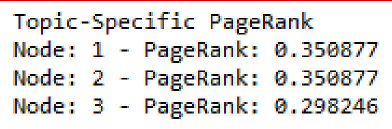
The webgraph is a directed graph, whose vertices correspond to the pages of the WWW, and a directed edge connects page X to page Y if there exists a hyperlink on page X, referring to page Y.

XML retrieval, or XML information retrieval, is the content-based retrieval of documents structured with XML (eXtensible Markup Language). As such it is used for computing relevance of XML documents.

C) CODE AND OUTPUT:

```
import networkx as nx
```

```
def parse_xml(xml_text):
    root = ET.fromstring(xml_text)
    return root


def generate_web_graph(xml_root):
    G = nx.DiGraph()


    for page in xml_root.findall('.//page'):
        page_id = page.find('id').text
        G.add_node(page_id)


        links = page.findall('.//link')
        for link in links:
            target_page_id = link.text
            G.add_edge(page_id,target_page_id)


    return G


def compute_topic_specific_pagerank(graph, topic_nodes, alpha=0.85, max_iter = 100, tol = 1e-6):
    personalization = {node: 1.0 if node in topic_nodes else 0.0 for node in graph.nodes}
    return nx.pagerank(graph, alpha=alpha, personalization=personalization, max_iter=max_iter, tol=tol)


if __name__ == "__main__":
    xml_data = """
    <webgraph>
        <page>
            <id>1</id>
            <link>2</link>

            <link>3</link>

        </page>

        <page>

            <id>2</id>

            <link>1</link>

            <link>3</link>

        </page>

        <page>

            <id>3</id>

            <link>1</link>

            <link>2</link>

        </page>

    </webgraph>"""

    xml_root = parse_xml(xml_data)

    web_graph = generate_web_graph(xml_root)

    topic_specific_pagerank = compute_topic_specific_pagerank(web_graph, topic_nodes=['1','2'])



    print("Topic-Specific PageRank")

    for node, score in sorted(topic_specific_pagerank.items(),key=lambda x:x[1], reverse=True):

        print(f"Node: {node} - PageRank: {score:4f}")
```



```
Topic-Specific PageRank
Node: 1 - PageRank: 0.350877
Node: 2 - PageRank: 0.350877
Node: 3 - PageRank: 0.298246
```

Name of Instructor: Mr.Rajesh Yadav

SIES

Information Retrieval

Practical No. 11

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Xml Retrieval	**Batch**	I
Date :	7/2/24	Practical No	11

A) AIM:

Write a python program to retrieve xml text using xml library.

B) DESCRIPTION:

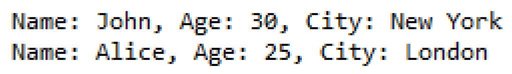
An Extensible Markup Language (XML) file is a text-based document that you can save with the .xml extension. You can write XML similar to other text files. To create or edit an XML file, you can use any of the following: Text editors like Notepad or Notepad++ Online XML editors.

XML retrieval, or XML information retrieval, is the content-based retrieval of documents structured with XML (eXtensible Markup Language). As such it is used for computing relevance of XML documents.

C) CODE AND OUTPUT:

```
import xml.etree.ElementTree as ET


xml_data = '''<root>

    <person>

        <name>John</name>

        <age>30</age>
```

```
            <city>New York</city>

        </person>

        <person>

            <name>Alice</name>

            <age>25</age>

            <city>London</city>

        </person>

    </root>'''


tree = ET.fromstring(xml_data)


for person in tree.findall('person'):

    name = person.find('name').text

    age = person.find('age').text

    city = person.find('city').text

    print(f"Name: {name}, Age: {age}, City: {city}")
```



```
Name: John, Age: 30, City: New York
Name: Alice, Age: 25, City: London
```

Information Retrieval

Practical No. 12

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS64	**Class**	B.Sc(Computer Science)
Topic:	Xml Retrieval	**Batch**	I
Date :	7/2/24	Practical No	12

A) AIM:

Write a python program to retrieve xml text using lxml library.

B) DESCRIPTION:

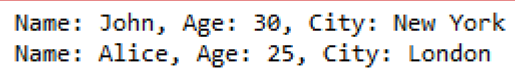
An Extensible Markup Language (XML) file is a text-based document that you can save with the .xml extension. You can write XML similar to other text files. To create or edit an XML file, you can use any of the following: Text editors like Notepad or Notepad++ Online XML editors.

XML retrieval, or XML information retrieval, is the content-based retrieval of documents structured with XML (eXtensible Markup Language). As such it is used for computing relevance of XML documents.

C) CODE AND OUTPUT:

```
from lxml import etree



xml_data = '''<root>

    <person>

        <name>John</name>

        <age>30</age>
```

```
            <city>New York</city>

        </person>

        <person>

            <name>Alice</name>

            <age>25</age>

            <city>London</city>

        </person>

    </root>'''


tree = etree.fromstring(xml_data)


for person in tree.xpath('//person'):

    name = person.xpath('name/text()')[0]

    age = person.xpath('age/text()')[0]

    city = person.xpath('city/text()')[0]

    print(f"Name: {name}, Age: {age}, City: {city}")
```



```
Name: John, Age: 30, City: New York
Name: Alice, Age: 25, City: London
```