

List of Experiments

Experiment No. 1

Task: Write a python program to implement Breadth First Search Traversal.

Solution:

```
graph = {
    '5' : ['3','7'],
    '3' : ['2', '4'],
    '7' : ['8'],
    '2' : [],
    '4' : ['8'],
    '8' : []
}


visited = [] # List for visited nodes.
queue = []    #Initialize a queue

def bfs(visited, graph, node): #function for BFS
    visited.append(node)
    queue.append(node)

    while queue:          # Creating loop to visit each node
        m = queue.pop(0)
        print (m, end = " ")

        for neighbour in graph[m]:
            if neighbour not in visited:
                visited.append(neighbour)
                queue.append(neighbour)

# Driver Code
print("Following is the Breadth-First Search")
bfs(visited, graph, '5')    # function calling
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====

Following is the Breadth-First Search

5 3 7 2 4 8

>>>

Experiment No. 2

Task: Write a python program to implement Water Jug Problem.

Solution:

```
from collections import defaultdict

jug1, jug2, aim = 4, 3, 2

visited = defaultdict(lambda: False)

def waterJugSolver(amt1, amt2):

    if (amt1 == aim and amt2 == 0) or (amt2 == aim and amt1 == 0):

        print(amt1, amt2)

        return True

    if visited[(amt1, amt2)] == False:

        print(amt1, amt2)

        visited[(amt1, amt2)] = True

        return (waterJugSolver(0, amt2) or

                waterJugSolver(amt1, 0) or

                waterJugSolver(jug1, amt2) or

                waterJugSolver(amt1, jug2) or

                waterJugSolver(amt1 + min(amt2, (jug1-amt1)),

                                amt2 - min(amt2, (jug1-amt1))) or

                waterJugSolver(amt1 - min(amt1, (jug2-amt2)),


                                amt2 + min(amt1, (jug2-amt2))))

    else:

        return False
```

```
print("Steps: ")
```

```
waterJugSolver(0, 0)
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====

Steps:

0 0

4 0

4 3

0 3

3 0

3 3

4 2

0 2

>>>

Experiment No. 3

Task: Write a python program to remove punctuations from the given string.

Solution:

```
# define punctuation

punctuations = ""!()-[]{};:'"\,.<>./?@$%^&*~_{}""

my_str = "Hello!!!, he said ---and went."

# To take input from the user

# my_str = input("Enter a string: ")

# remove punctuation from the string


no_punct = ""

for char in my_str:

    if char not in punctuations:

        no_punct+=char

print(no_punct)
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====
```

```
Hello he said and went
```

```
>>> |
```

Experiment No. 4

Task: Write a python program to sort the sentence in alphabetical order.

Solution:

```
my_str = input("Enter a string: ")

# breakdown the string into a list of words

words = my_str.split()


# sort the list

words.sort()

# display the sorted words

for word in words:

    print(word, end=" ")
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====
Enter a string: This is a sentence.
Thisaisentence.
>>> |
```

Experiment No. 5

Task: Write a program to implement Hangman game using python.

Solution:

```
#importing the time module

import time

import random

#welcoming the user

name = input("What is your name? ")

print ("Hello, " + name, "Time to play hangman!")

#wait for 1 second

time.sleep(1)

print ("Start guessing...")

time.sleep(0.5)

#here we set the secret. You can select any word to play with.

words = ['rainbow', 'computer', 'science', 'programming',
         'python', 'mathematics', 'player', 'condition',
         'reverse', 'water', 'board', 'geeks']

word = random.choice(words)

#creates an variable with an empty value

guesses = ""

#determine the number of turns

turns = 10

# Create a while loop

#check if the turns are more than zero
```

```
while turns > 0:

    # make a counter that starts with zero

    failed = 0

    # for every character in secret_word

    for char in word:

        # see if the character is in the players guess

        if char in guesses:

            # print then out the character

            print (char,end=""),

        else:

            # if not found, print a dash

            print ("_",end=""),

        # and increase the failed counter with one

        failed += 1

    # if failed is equal to zero

    # print You Won

    if failed == 0:

        print ("\nYou won")

    # exit the script

    break

    # ask the user go guess a character

    guess = input("\nguess a character:")

    # set the players guess to guesses

    guesses += guess
```

```
# if the guess is not found in the secret word
```

```
if guess not in word:
```

```
# turns counter decreases with 1 (now 9)
```

```
    turns -= 1
```

```
# print wrong
```

```
    print ("Wrong")
```

```
# how many turns are left
```

```
    print ("You have", + turns, 'more guesses' )
```

```
# if the turns are equal to zero
```

```
    if turns == 0:
```

```
# print "You Lose"
```

```
        print ("You Lose" )
```

```
IDLE Shell 3.10.6
File Edit Shell Debug Options Window Help
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====
What is your name? Bhaskar
Hello, Bhaskar Time to play hangman!
Start guessing...

guess a character:w
Wrong
You have 9 more guesses

guess a character:s
Wrong
You have 8 more guesses

guess a character:b
b__
guess a character:o
bo__
guess a character:a
boa__
guess a character:r
boar_
guess a character:d
board
You won
>>> |
```


Experiment No. 6

Task: Write a program to implement Tic-Tac-Toe game using python.

Solution:

```
import random

class TicTacToe:

    def __init__(self):

        self.board = []

    def create_board(self):

        for i in range(3):

            row = []

            for j in range(3):

                row.append('-')

            self.board.append(row)

    def get_random_first_player(self):

        return random.randint(0, 1)

    def fix_spot(self, row, col, player):

        self.board[row][col] = player

    def is_player_win(self, player):

        win = None

        n = len(self.board)

        # checking rows
        for i in range(n):

            win = True

            for j in range(n):

                if self.board[i][j] != player:

                    win = False
```

```
        break

    if win:

        return win

# checking columns
for i in range(n):

    win = True

    for j in range(n):

        if self.board[j][i] != player:

            win = False

            break

    if win:

        return win

# checking diagonals
win = True

for i in range(n):

    if self.board[i][i] != player:

        win = False

        break

if win:

    return win

win = True

for i in range(n):

    if self.board[i][n - 1 - i] != player:

        win = False

        break

if win:
```

```

        return win

    return False

    for row in self.board:

        for item in row:

            if item == '-':

                return False

    return True

def is_board_filled(self):

    for row in self.board:

        for item in row:

            if item == '-':

                return False

    return True

def swap_player_turn(self, player):

    return 'X' if player == 'O' else 'O'

def show_board(self):

    for row in self.board:

        for item in row:

            print(item, end=" ")

        print()

def start(self):

    self.create_board()

    player = 'X' if self.get_random_first_player() == 1 else 'O'

    while True:

        print(f"Player {player} turn")

        self.show_board()

```

```

# taking user input

row, col = list(
    map(int, input("Enter row and column numbers to fix spot: ").split()))

print()

# fixing the spot

self.fix_spot(row - 1, col - 1, player)

# checking whether current player is won or not

if self.is_player_win(player):

    print(f"Player {player} wins the game!")

    break

# checking whether the game is draw or not

if self.is_board_filled():

    print("Match Draw!")

    break

# swapping the turn

player = self.swap_player_turn(player)

# showing the final view of board

print()

self.show_board()

# starting the game

tic_tac_toe = TicTacToe()

tic_tac_toe.start()

```

Python Shell 3.10.6

File Edit Shell Debug Options Window Help

Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====

Player X turn

- - -
- - -
- - -

Enter row and column numbers to fix spot: 1 1

Player O turn

X - -
- - -
- - -

Enter row and column numbers to fix spot: 2 2

Player X turn

X - -
- O -
- - -

Enter row and column numbers to fix spot: 1 3

Player O turn

X - X
- O -
- - -

Enter row and column numbers to fix spot: 1 2

Player X turn

X O X
- O -
- - -

Enter row and column numbers to fix spot: 3 1

Player O turn

X O X
- O -
X - -

Enter row and column numbers to fix spot: 3 2

Player O wins the game!

X O X
- O -
X O -

>>> |

Experiment No. 7

Task: Write a python program to remove stop words for a given passage from a text file using NLTK.

Solution:

```
from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

example_sent = """This is a sample 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.lower() 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)
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====

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

['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration', '.']

>>>

Experiment No. 8

Task: Write a python program to implement stemming for a given sentence using NLTK.


Solution:

```
# importing modules
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

sentence = "Programmers program with programming languages"
words = word_tokenize(sentence)

for w in words:
    print(w, " : ", ps.stem(w))
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====

Programmers : programm

program : program

with : with

programming : program

languages : languag

>>>

Experiment No. 9

Task: Write a python program to POS (Parts of Speech) tagging for the give sentence using NLTK.

Solution:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stop_words = set(stopwords.words('english'))

# Dummy text
txt = "Sukanya, Rajib and Naba are my good friends. "
"Sukanya is getting married next year. "
"Marriage is a big step in one's life."
"It is both exciting and frightening. "
"But friendship is a sacred bond between people."
"It is a special kind of love between us. "
"Many of you must have tried searching for a friend "
    "but never found the right one."

# sent_tokenize is one of instances of
# PunktSentenceTokenizer from the nltk.tokenize.punkt module
tokenized = sent_tokenize(txt)
for i in tokenized:

    # Word tokenizers is used to find the words
    # and punctuation in a string
    wordsList = nltk.word_tokenize(i)

    # removing stop words from wordList
    wordsList = [w for w in wordsList if not w in stop_words]

    # Using a Tagger. Which is part-of-speech
    # tagger or POS-tagger.
    tagged = nltk.pos_tag(wordsList)
    print(tagged)
```

```
IDLE Shell 3.10.6
File Edit Shell Debug Options Window Help
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====
[('Sukanya', 'NNP'), (',', ','), ('Rajib', 'NNP'), ('Naba', 'NNP'), ('good', 'JJ'), ('friends', 'NNS'), ('.', '.')]
>>>
```


Experiment No. 10

Task: Write a python program to implement Lemmatization using NLTK.


Solution:

```
# import these modules
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print("rocks :", lemmatizer.lemmatize("rocks"))
print("corpora :", lemmatizer.lemmatize("corpora"))

# a denotes adjective in "pos"
print("better :", lemmatizer.lemmatize("better", pos="a"))
```

 IDLE Shell 3.10.6

File Edit Shell Debug Options Window Help

Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:/Users/Bhaskar Banerjee/OneDrive/Desktop/AI_lab.py =====

rocks : rock

corpora : corpus

better : good

>>> |