**REPORT 7**



수강과목 : 시스템성능분석

담당교수 : 서성현 교수님

학당교과 : 컴퓨터정보공학부

학당과년 : 4학년

학당과번 : 201921285

이당과름 : 정소희

 제출일자 : 2022.06.03

<8장 응용 예제 1>

- 전체 코드

import random

class TreeNode():

    def \_\_init\_\_ (self):

        self.left = None

        self.data = None

        self.right = None

memory = []

root = None

dataAry = ['바나나맛우유','레쓰비캔커피','츄파츕스','도시락','삼다수','코카콜라','삼각김밥']

sellAry = [random.choice(dataAry) for \_ in range(20)]

print('오늘 판매된 물건(중복O) --> ', sellAry)

node = TreeNode()

node.data = sellAry[0]

root = node

memory.append(node)

for name in sellAry[1:]:

    node = TreeNode()

    node.data = name

    current = root

    while True:

        if name == current.data:

            break

        if name < current.data:

            if current.left == None:

                current.left = node

                memory.append(node)

                break

            current = current.left

        else:

            if current.right == None:

                current.right = node

                memory.append(node)

                break

            current = current.right

print("이진 탐색 트리 구성 완료!")

def preorder(node):

    if node == None:

        return

    print(node.data, end = ' ')

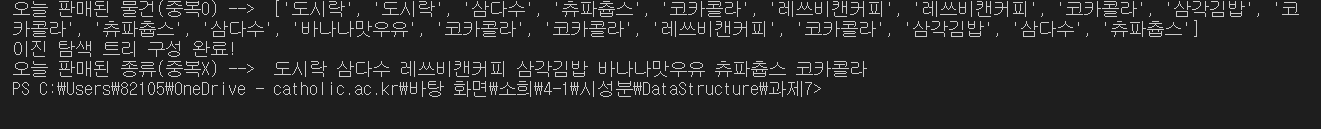
    preorder(node.left)

    preorder(node.right)

print('오늘 판매된 종류(중복X) --> ',end =' ')

preorder(root)

- 실행 화면 캡처



<8장 응용 예제2>

- 전체 코드

import os

class TreeNode():

    def \_\_init\_\_ (self):

        self.left = None

        self.data = None

        self.right = None

memory = []

root = None

fnameAry =[]

folderName = 'C:/Users/82105/OneDrive - catholic.ac.kr/바탕 화면/소희/4-1/시성분/DataStructure'

for dirName, subDirList, fnames in os.walk(folderName) :

    for fname in fnames:

        fnameAry.append(fname)

node = TreeNode()

node.data = fnameAry[0]

root = node

memory.append(node)

dupNameAry = []

for name in fnameAry[1:]:

    node = TreeNode()

    node.data = name

    current = root

    while True:

        if name == current.data:

            dupNameAry.append(name)

            break

        if name <current.data:

            if current.left == None:

                current.left = node

                memory.append(node)

                break

            current = current.left

        else:

            if current.right == None:

                current.right = node

                memory.append(node)

                break

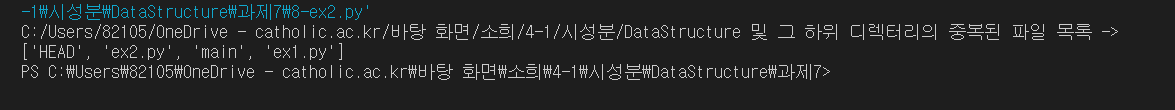
            current = current.right

dupNameAry = list(set(dupNameAry))

print(folderName,'및 그 하위 디렉터리의 중복된 파일 목록 -> ')

print(dupNameAry)

- 코드 실행 화면 캡처



<9장 응용 예제 1>

- 전체 코드

class Graph():

    def \_\_init\_\_ (self, size):

        self.SIZE = size

        self.graph = [[0 for \_ in range(size)] for \_ in range(size)]

def printGraph(g):

    print(' ', end =' ')

    for v in range(g.SIZE) :

        print("%9s" % storeAry[v][0], end= ' ')

    print()

    for row in range(g.SIZE):

        print("%9s" % storeAry[row][0], end= ' ')

        for col in range(g.SIZE):

            print("%8d" % g.graph[row][col], end =' ')

        print()

    print()

G1 = None

storeAry = [['GS25',30],['CU',60],['Seven11',10],['MiniStop',90],['Emart24',40]]

GS25,CU,Seven11,MiniStop,Emart24 = 0,1,2,3,4

gSize = 5

G1 = Graph(gSize)

G1.graph[GS25][CU] = 1; G1.graph[GS25][Seven11] =1;

G1.graph[CU][GS25] = 1; G1.graph[CU][Seven11] =1; G1.graph[CU][MiniStop] =1;

G1.graph[Seven11][GS25] = 1; G1.graph[Seven11][CU] =1; G1.graph[Seven11][MiniStop] =1;

G1.graph[MiniStop][Emart24] = 1; G1.graph[MiniStop][CU] =1; G1.graph[MiniStop][Seven11] =1;

G1.graph[Emart24][MiniStop] =1;

print('#편의점 그래프')

printGraph(G1)

stack = []

visitedAry = []

current =0

maxStore = current

maxCount = storeAry[current][1]

stack.append(current)

visitedAry.append(current)

while(len(stack)!=0) :

    next = None

    for vertex in range(gSize):

        if G1.graph[current][vertex] ==1:

            if vertex in visitedAry:

                pass

            else:

                next = vertex

                break

    if next != None:

        current = next

        stack.append(current)

        visitedAry.append(current)

        if storeAry[current][1]>maxCount:

            maxCount = storeAry[current][1]

            maxStore = current

    else:

        current = stack.pop()

print('허니버터칩 최대 보유 편의점(개수) --> ', storeAry[maxStore][0],'(',storeAry[maxStore][1],')')

- 실행 화면 캡처

텍스트이(가) 표시된 사진

자동 생성된 설명

<9장 응용 예제2>

- 전체 코드

class Graph():

    def \_\_init\_\_ (self, size):

        self.SIZE = size

        self.graph = [[0 for \_ in range(size)] for \_ in range(size)]

def printGraph(g):

    print(' ', end =' ')

    for v in range(g.SIZE) :

        print(cityAry[v], end= ' ')

    print()

    for row in range(g.SIZE):

        print(cityAry[row], end= ' ')

        for col in range(g.SIZE):

            print("%2d" % g.graph[row][col], end =' ')

        print()

    print()

def findVertex(g, findVtx):

    stack = []

    visitedAry = []

    current =0

    stack.append(current)

    visitedAry.append(current)

    while(len(stack)!=0) :

        next = None

        for vertex in range(gSize):

            if g.graph[current][vertex] ==1:

                if vertex in visitedAry:

                    pass

                else:

                    next = vertex

                    break

        if next != None:

            current = next

            stack.append(current)

            visitedAry.append(current)

        else:

            current = stack.pop()

    if findVtx in visitedAry:

        return True

    else:

        return False

G1 = None

cityAry = ['서울','뉴욕','런던','북경','방콕','파리']

서울,뉴욕,런던,북경,방콕,파리 = 0,1,2,3,4,5

gSize = 6

G1 = Graph(gSize)

G1.graph[서울][뉴욕] =80; G1.graph[서울][북경] = 10;

G1.graph[뉴욕][서울] =80; G1.graph[뉴욕][북경] =40; G1.graph[뉴욕][방콕] =70;

G1.graph[런던][파리] =60; G1.graph[런던][방콕] =30;

G1.graph[북경][서울] =10; G1.graph[북경] [뉴욕]=40; G1.graph[북경][방콕] =50;

G1.graph[방콕][런던] =30; G1.graph[방콕] [뉴욕]=70; G1.graph[방콕][북경] =50; G1.graph[방콕][파리] =20;

G1.graph[파리][방콕] =20;G1.graph[파리][런던] =60;

print('#해저 케이블 연결을 위한 전체 연결도#')

printGraph(G1)

from operator import itemgetter

edgeAry = []

for i in range(gSize):

    for k in range(gSize):

        if G1.graph[i][k] !=0:

            edgeAry.append([G1.graph[i][k],i,k])

edgeAry = sorted(edgeAry, key=itemgetter(0), reverse=False)

newAry = []

for i in range(0,len(edgeAry), 2):

    newAry.append(edgeAry[i])

index = 0

while (len(newAry) > gSize-1 ):

    start = newAry[index][1]

    end = newAry[index][2]

    saveCost = newAry[index][0]

    G1.graph[start][end] = 0

    G1.graph[end][start] = 0

    startYN = findVertex(G1,start)

    endYN = findVertex(G1, end)

    if startYN and endYN:

        del(newAry[index])

    else:

        G1.graph[start][end] = saveCost

        G1.graph[end][start] = saveCost

        index +=1

print('가장 효율적인 해저 케이블 연결도')

printGraph(G1)

- 코드 실행 화면 캡처

텍스트이(가) 표시된 사진

자동 생성된 설명

<10장 응용 예제 1>

- 전체 코드

def notation(base, n):

    if n <base :

        print(numberChar[n], end = ' ')

    else:

        notation(base, n //base)

        print(numberChar[n%base], end =' ')

numberChar = ['0','1','2','3','4','5','6','7','8','9']

numberChar += ['A','B','C','D','E','F']

number = int(input('10진수 입력 --> '))

print('\n2진수 :  ',end = ' ')

notation(2,number)

print('\n8진수 :  ',end = ' ')

notation(8,number)

print('\n16진수 :  ',end = ' ')

notation(16,number)

- 실행 화면 캡처

텍스트이(가) 표시된 사진

자동 생성된 설명

<10장 응용 예제2>

- 전체 코드

from tkinter import \*

window = Tk()

window.title("삼각형 프랙탈")

def drawTriangle(x,y,size):

    if size >= 30:

        drawTriangle(x,y,size/2)

        drawTriangle(x+size/2,y,size/2)

        drawTriangle(x+size/4, int(y-size\*(3\*\*0.5)/4), size/2)

    else:

        canvas.create\_polygon(x, y, x+size, y, x+size/2, y-size\*(3\*\*0.5)/2, fill="red", outline="red")

wSize = 500

radius = 200

canvas = Canvas(window, height=wSize, width=wSize, bg='white')

drawTriangle(wSize/5, wSize/5\*4, wSize\*2/3)

canvas.pack()

window.mainloop()

- 코드 실행 화면 캡처

