Assignment 01

LinQiangMa

October 2023

Problem1 Flowchart

```
import random
a, b, c = random.random()*200-100, random.random()*200-100, random.random()*200-100
print("The  values of a, b, and c:")
print("a= ", a)
print("b= ", b)
print("c= ", c,end="\n\n")

if a > b:
    if b > c:
        print("The order of abc from largest to smallest is: \na, b, c")
    elif a > c:
        print("The order of abc from largest to smallest is: \na, c, b")
    else:
        print("The order of abc from largest to smallest is: \nc, a, b")
else:
    if a > c:
        print("The order of abc from largest to smallest is: \nb, a, c")
    elif b > c:
        print("The order of abc from largest to smallest is: \nb, c, a")
    else:
        print("The order of abc from largest to smallest is: \nc, b, a")
del a, b, c

output:
The  values of a, b, and c:
a=  -20.031534406037892
b=  -5.134063588278394
c=  45.864282110488574

The order of abc from largest to smallest is:
c, b, a
```

Problem2 Matrix multiplication

```python
import random
#build matrices
row, column = 5, 10
a = []
for i in range(row):
    b = []
    for j in range(column):
        b.append(random.randint(0,50))
    a.append(b)
    del b

row, column = 10,5
c=[]
for i in range(row):
    b=[]
    for j in range(column):
        b.append(random.randint(0,50))
    c.append(b)
    del b

#do matrix multiplication
def matrix_multip(a, b):
    if len(a[0]) != len(b):
        print("The matrix multiplication is not possible.")
        return 0

    else:
        row = len(a)
        column = len(b[0])
        result = []
        for i in range(row):
            result.append([])
            for j in range(column):
                result[i].append(0)
                for k in range(len(b)):
                    result[i][j] = result[i][j] + a[i][k] * b[k][j]
        return result

#output
print("The first matrix is:")
for row in a:
    print(row)
print(" ")
print("The second matrix is:")
for row in c:
    print(row)
print(" ")
if matrix_multip(a, c) == 0:
   pass
else:
    print("The result of matrix multiplication is:")
    for row in matrix_multip(a, c):
        print(row)
```

```
output:
The first matrix is:
[46, 15, 34, 7, 38, 14, 2, 48, 27, 12]
[40, 40, 6, 17, 14, 42, 27, 20, 2, 43]
[36, 30, 14, 25, 10, 27, 18, 41, 39, 12]
[34, 40, 3, 30, 7, 27, 47, 8, 20, 48]
[15, 28, 24, 35, 44, 3, 15, 23, 14, 18]

The second matrix is:
[47, 19, 24, 21, 25]
[8, 24, 37, 9, 8]
[31, 2, 18, 0, 9]
[18, 14, 12, 40, 2]
[40, 13, 20, 43, 50]
[29, 7, 26, 37, 39]
[14, 21, 22, 26, 11]
[22, 41, 3, 3, 42]
[18, 16, 29, 45, 8]
[21, 28, 9, 26, 5]

The result of matrix multiplication is:
[7210, 4770, 4558, 5256, 6350]
[6227, 5069, 5223, 6006, 5114]
[6107, 5120, 5186, 6113, 5161]
[5816, 5291, 5622, 6968, 3913]
[5496, 4074, 4173, 5527, 4535]
```

Problem3 Pascal triangle

```
#Junleizhou and Junhongcai explained to me what is recursion algorithm.
def Pascal_triangle(n):
    if n == 1:
        return [1]
    else:
        result = [1]
        lastline = Pascal_triangle(n-1)
        for i in range(1, len(lastline)):
            result.append(lastline[i-1]+lastline[i])
        result.append(1)
        return result
print("The 100th line of Pascal's triangle is:", Pascal_triangle(100), end="\n\n")
print("The 200th line of Pascal's triangle is:", Pascal_triangle(200))

output:
The 100th line of Pascal's triangle is: [1, 99, 4851, 156849,...,1]
The 200th line of Pascal's triangle is: [1, 199, 19701, 1293699, ...,1]
```

Problem4 Add or double

```
#Junleizhou and Junhongcai explained to me what is recursion algorithm
import random
def Least_moves(n):
    if n == 1:
        return 0
    if n % 2 == 0:
        return 1 + Least_moves(n/2)
    else:
        return 1 + Least_moves(n-1)
b = random.randint(1,100)
print("The smallest number of moves from 1 to ",b,"is:",Least_moves(b))

output:
The smallest number of moves from 1 to  32 is: 5
```

Problem5 Dynamic programming

```
#Junleizhou, Junhongcai and Zhouzhou explained to me what is recursion and inspired me
#to solve the problem.
import random
import numpy as np
def Find_expression(N):
    n=0 #record the number of expression
    list=[] #record the every expression
    def NEXT(a,b,end):
        nonlocal n
        if b>9:
           if eval(a)==end:
            list.append(a)
            n=n+1
        else:
            NEXT(str(a)+"+"+str(b),b+1,end)
            NEXT(str(a)+"-"+str(b),b+1,end)
            NEXT(str(a)+str(b),b+1,end)
    NEXT(1,2,N)
    return [n,list]
n=random.randint(1,101)
print("The nymber of expressions which equal",n,"is",Find_expression(n)[0],"\nThey are:")
for i in Find_expression(n)[1]:
    print(i,"=",n)
for i in range(100):
    if i == 0:
        T=[]
        T.append(Find_expression(i+1)[0])
    else:
        T.append(Find_expression(i+1)[0])#record the number of expression that equal i+1
T=np.array(T)
print("The maximum number of expressions is",np.max(T),"when n is")
for i in np.where(T==np.max(T))[0]:
    print (i+1)
print("The minimum number of expressions is",np.min(T),"when n is")
for i in np.where(T==np.min(T))[0]:
    print (i+1)

output:
The nymber of expressions which equal 98 is 9
They are:
1+2+3+4-5+6+78+9 = 98
1+2+34-5+67+8-9 = 98
1+23+4-5+6+78-9 = 98
1-2-3+4+5+6+78+9 = 98
1-23+45+6+78-9 = 98
12+3+4-5+67+8+9 = 98
12-3+4-5-6+7+89 = 98
12-3-4+5+6-7+89 = 98
123-4-5-6+7-8-9 = 98
The maximum number of expressions is 26 when n is
1
45
The minimum number of expressions is 6 when n is
88
```