과목 명: 시스템프로그래밍

담당 교수 명: 소 정 민

<<Assignment 4>>

**서강대학교 컴퓨터공학과**

**20171601**

**강주형**

**목 차**

1. 프로그램 개요 3

2. 프로그램 설명 3

3. 모듈 정의 3

3.1 모듈 이름 : p1.ipynb 3

3.1.1 알고리즘 설명 3

3.2 모듈 이름: p2.ipynb 3

3.2.1 알고리즘 설명 3

3.3 모듈 이름: p3.ipynb 3

3.3.1 알고리즘 설명 3

3.4 모듈 이름: p4.ipynb 3

3.4.1 알고리즘 설명 3

4. 코드 4

4.1 p1.ipynb 4

4.2 p2.ipynb 4

4.3 p3.ipynb 5

4.4 p4.ipynb 6

1. **프로그램 개요**

이 프로젝트는 파이썬 프로그래밍에 대한 이해를 높이기 위한 것을 목표로

하여 파이썬을 이용하여 4가지 프로그램을 짜보는 것이 과제이다.

1. **프로그램 설명**

첫 번째 과제는 주어진 배열에 따라 counter의 값을 정하는 것이다. 두 번째 과제는 주어진 두 개의 배열에 있는 숫자의 소인수가 서로 동일한지 판단하는 것이다. 세 번째 과제는 주어진 N개의 숫자들의 조합 중에서 가장 작은 합의 절댓값을 구하는 것이다. 마지막 과제는 이차원 배열에서 2x2 배열의 가장 최솟값을 찾아 원래 배열의 크기를 반으로 줄이는 것이다.

1. **모듈 정의**
   1. **모듈 이름 : py1.ipynb**
      1. *알고리즘 설명*

처음에 카운터의 개수와 Array를 입력으로 받는다. 그 후에 for문을 돌면서 만약 Array의 원소가 1보다 크거나 같고 N보다 작거나 같다면 그 Array 값의 인덱스에 해당하는 counter 원소의 값을 1만큼 증가시키고 만약 Array의 원소가 N+1의 값과 같다면 counter의 원소들 중에서 최댓값을 찾아 모든 counter의 값을 그 최댓값으로 초기화시킨다. 그렇게 Array의 개수만큼 반복문을 돌면 결과물을 출력한다.

* 1. **모듈 이름 : py2.ipynb**
     1. *알고리즘 설명*

시작할 때 Array A와 Array B를 입력 받는다. 그리고 Array의 개수만큼 반복문을 돌면서 첫 번째로는 index i에 해당하는 원소의 약수를 구한다. 그 값들을 새로운 set에 각각 넣는다. 약수들을 다 찾았다면 다시 반복문을 돌면서 소수의 집합인 소인수를 찾아낸다. 그리고 소인수들을 또 다시 새로운 set에 각각 넣는다. 그래서 두 set을 비교해서 만약 두 set의 원소가 같다면, 즉 소인수가 같다면 count를 1 증가해준다.

* 1. **모듈 이름 : py3.ipynb**
     1. *알고리즘 설명*

시작할 때 Array를 입력 받는다. 그런 후에 반복문을 돌면서 모든 인덱스를 방문하여 두 수의 합이 음수라면 –1을 곱해 양수를 변환 한 후 처음에 지정한 minimum값과 비교하여 더 작은 값을 새롭게 minimum값으로 설정해준다. Array의 개수만큼 반복문을 돈 후 마지막에 저장된 minimum값을 결과물로 출력한다.

* 1. **모듈 이름 : py4.ipynb**
     1. *알고리즘 설명*

시작할 때, 배열의 크기를 입력 받는다. 그리고 입력 받은 크기를 바탕으로 랜덤 배열을 생성한다. 이 알고리즘의 목적은 배열의 사이즈를 반으로 줄이는 것이기 때문에 입력 받을 배열의 크기를 짝수라고 가정한다. 랜덤 생성한 배열을 2X2 배열 조각으로 나눠서 2x2 배열의 4개의 원소 중 가장 큰 원소를 새로운 리스트 result에 저장한다. 그리고 이 리스트를 크기가 (N/2)x(N/2)인 배열로 변형한 후 그 결과물을 출력한다.

1. **코드**

**4.1**  **p1.ipynb**

print("Input:")

N = int(input()) #counter 갯수 입력

A = list(input().split(" ")) #Array 입력

A = [int (i) for i in A]

counter = [0] \* N #counter 초기화

for i in range(len(A)):

if (A[i] >= 1 and A[i] <= N): #Array의 숫자가 1보다 크고 N보다 작으면

index = A[i] #숫자에 해당하는 인덱스에

counter[index-1] += 1 #해당하는 counter를 1증가

elif (A[i] == N+1): #Array의 숫자가 N+1과 같으면

maxValue = counter[0] #counter의 최댓값을 찾아서

for k in range(1, len(counter)):

if maxValue < counter[k]:

maxValue = counter[k]

counter = [maxValue]\*N #그 최댓값으로 모든 counter를 set해줌

else:

continue

print()

print("Output:")

for i in counter:

print(i,end=" ")

**4.2**  **p2.ipynb**

print("Input:")

A = list(input().split(" ")) #Array A 입력

A = [int (i) for i in A]

B = list(input().split(" ")) #Array B 입력

B = [int (i) for i in B]

count=0

for i in range(len(A)):

a = set()

b = set()

for k in range(A[i]-1): #A[i]의 약수 찾기

if (A[i] % (k+2) == 0 ):

a.add(k+2)

else:

continue

for k in range(B[i]-1): #B[i]의 약수 찾기

if (B[i] % (k+2) == 0):

b.add(k+2)

else:

continue

a=list(a)

b=list(b)

c=set()

d=set()

for i in a: #A[i]의 소인수 찾기

for j in a:

if(i == j):

continue

if(i%j == 0):

c.add(i)

else:

continue

for i in b: #B[i]의 소인수 찾기

for j in b:

if(i == j):

continue

if(i%j == 0):

d.add(i)

if(i == j):

continue

a=set(a)

b=set(b)

c=a-c

d=b-d

if (c == d): #A[i]와 B[i]의 소인수가 같으면 count 1증가

count += 1

print()

print("Output: ")

print(count)

**4.3**  **p3.ipynb**

print("Input:")

A = list(input().split(" ")) #Array 입력

A = [int (i) for i in A]

if((A[0]+A[0])<0): #minumum값 지정

min = (-1)\*(A[0]+A[0])

else:

min = A[0]+A[0]

for i in range(len(A)):

for j in range(i,len(A)):

if((A[i]+A[j])<0): #만약 더한 값이 음수라면 양수로 변환

result = (-1)\*(A[i]+A[j])

else:

result = A[i]+A[j]

if(result<min):

min = result #더한 값이 minimum 값보다 작으면 그 값을 minimum으로 새로 설정

print()

print("Output:")

print(min)

**4.4**  **p4.ipynb**

import numpy as np

print("Input:")

N = int(input()) #배열의 크기 입력

A = np.random.random((N,N)) #랜덤 배열 생성

result = []

for i in range(0,N,2): #2x2배열로 나눠서

for j in range(0,N,2):

max = A[i][j] #최댓값 초기 설정

if(A[i][j+1] > max): #최댓값 찾기

max = A[i][j+1]

if(A[i+1][j] > max):

max = A[i+1][j]

if(A[i+1][j+1] > max):

max = A[i+1][j+1]

result.append(max)

B = np.array(result) #결과값이 든 리스트를 배열로 변경

B = B.reshape(N//2,N//2) #2x2 배열로 바꾸기

print()

print("Output:")

print(A)

print(B)