과목 명: 시스템프로그래밍

담당 교수 명: 소 정 민

<<Assignment 4>>

**서강대학교 컴퓨터공학과**

**[학번 : 20141515]**

**[이름 : 김연후]**

목 차

1. 프로그램 개요 3

2. 프로그램 설명 3

3. 모듈 정의 3

3.1 Max Counters 3

3.2 Common Prime divisors 3

3.3 Min abs sum of two 4

3.4 Max pooling 4

4. 전역 변수 정의 4

5. 코드 4-5

# 프로그램 개요

모든 코드들은 Python3로 작성되었으며, 작업환경은 Google의 Colaboratory이다.

Input은 user로부터 입력 받고, Output은 screen에 출력되어 나온다.

# 프로그램 설명

1번 : Counters를 구하는 문제다. Input으로 N (counter의 개수), Array A(counter를 조건에 따라 증가시킴)가 주어지고, Output은 Counter의 결과를 출력한다.

N개의 counter는 0으로 초기화 된다. Counter를 바꾸는 조건은 다음과 같다.

만약, A[K] = X (1≤ X ≤ N) 이면, counter X를 1 증가 시킨다.

만약, A[K] = N+1 이면, 모든 counter값들을 현재 최댓값을 갖는 counter값으로 바꾼다.

2번 : Common Prime divisors를 구하는 문제다. Input으로 Array A, Array B가 입력되고, Output은 A와B의 Common Prime divisor set가 같은 경우의 개수를 출력한다. Common Prime divisor set의 비교는 A[K], B[K] 두 수의 prime divisor를 비교한다.

3번 : Array A의 항목 중 중복을 가능하게 하여 2개를 선택하여 합의 절댓값 중 최솟값을 찾는 문제다. Input으로 Array A를 받고, Output으로 최솟값을 출력한다.

4번 : 2차원numpy array(matrix)를 절반 크기로 압축시키는 것이다.(ex. 4x4 -> 2x2)

이 때 압축 시, 각각의 2x2 영역 (6x6의 경우 9개의 2x2의 영역 존재)에서 최댓값을 압축된 matrix에 넣는다(Max pooling). Input으로 square matrix size를 받고, Output으로 randomly generated numpy array와 result of max pooling about array를 출력한다.

# 모듈 정의

## Max Counters

for문을 이용하여 Counter들을 0으로 초기화하였고, for문을 이용하여 입력된 Array A의 항목들에 대해 if문으로 N보다 큰 경우는 모든 counter들을 현재 최댓값을 갖는 counter 값으로 바꿔주고, 그렇지 않은 경우는 해당 Array A 항목 값을 번호로 갖는 counter를 1 증가 시켰다.

## Common prime divisors

isPrime(a) 이라는 함수를 정의하여 a가 소수인지 판별하는 함수를 만들었고, 이를 이용하여 임의의 숫자까지의 소수들을 모두 갖는 리스트를 만든다. 이 리스트를 이용하여, A[i]와 B[i]의 prime divisor set를 구한다. if문을 이용하여 A[i]와 B[i]가 소수들을 갖는 리스트의 소수들로 나뉠 경우 set에 포함, 나뉘지 않는 경우 제외한다.

## Min abs sum of two

입력을 받아 이중 for문으로 하나하나 더해봐서 합의 절댓값이 최솟값을 갖는 pair를 찾아서 그 값을 출력한다.

## Max pooling

import numpy as np 로 numpy를 불러와 np로 표시하고, np.random 함수를 이용하여 입력받은 size 만큼의 size\*size크기의 이차원 random matrix를 만들고 이중 for문을 이용하여 random matrix의 각 2x2 부분마다의 최댓값을 np.max 함수를 이용하여 구하고, 새로운 행렬에 값을 넣어 max pooling 해준다.

# 전역 변수 정의

없음(x)

# 코드

1. Max Counters

N = int(input())

str = input()

A = str.split()

L = []

for i in range(N):

L.append(0)

for i in range(len(A)):

if(int(A[i])<=N):

L[int(A[i])-1]+=1

else:

for j in range(0,len(L)):

L[j]=max(L)

for i in range(N):

print(L[i], end =' ')

1. Common prime divisors

def isPrime(a):

if(a<2):

return False

for i in range(2,a):

if(a%i==0):

return False

return True

A = input().split()

B = input().split()

result = 0

for i in range(len(A)):

prime=[]

pdA=[]

N = int(A[i])

for j in range(1,N+1):

if(isPrime(j) == True):

prime.append(j)

for j in range(len(prime)):

if(int(A[i])%prime[j] == 0):

pdA.append(prime[j])

pdA = set(pdA)

prime=[]

pdB=[]

N = int(B[i])

for j in range(1,N+1):

if(isPrime(j) == True):

prime.append(j)

for j in range(len(prime)):

if(int(B[i])%prime[j] == 0):

pdB.append(prime[j])

pdB = set(pdB)

if(pdA == pdB):

result += 1

print(result)

1. Min abs sum of two

A = input().split()

for i in range(len(A)):

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

v = abs(int(A[i])+int(A[j]))

if(i==0 and j==0):

m = v

if(m > v):

m = v

print(m)

1. Max pooling

import numpy as np

size = int(input())

A = np.random.random((size,size))

print(A)

n = int(size/2)

B = np.random.random((n,n))

for i in range(0,n):

for j in range(0,n):

B[i,j] = np.max(A[2\*i:2\*i+2,2\*j:2\*j+2])

print(B)