

Description of coding

No.4

문제 1

주어진 텍스트 파일의 경로를 읽어서 길이가 짧은 단어부터(길이가 같으면 사전 순서) 정렬하여 알파벳 소문자로만 출력해주세요.(텍스트 파일은 알파벳으로만 작성, 텍스트 파일은 1메가 이하)

예)

입력:

/home/nara/test.txt

출력:

I
im
in
no
but
more
must
cannot

No.4

```
class Nsort:
    def __init__(self, li, sep):
        self.li = li.split(sep)
        self.li = {i.strip().lower():len(i.strip()) for i in self.li}
        self.li = sorted(self.li.items(), key = operator.itemgetter(1))
        self.mBox = list()
        self.eqBox = list()

    def eqSort(self):
        list(map(lambda x : self.mBox.append(x), sorted(self.eqBox)))
        self.eqBox = list()

    def con(self):
        for i in range(len(self.li)):
            if i != len(self.li)-1:
                if self.li[i][1] != self.li[i-1][1] and self.li[i][1] != self.li[i+1][1]:
                    self.mBox.append(self.li[i])

                if self.li[i][1] == self.li[i-1][1] or self.li[i][1] == self.li[i+1][1]:
                    self.eqBox.append(self.li[i])

                if self.li[i][1] != self.li[i+1][1]:
                    self.eqSort()

        if self.eqBox != []:
            self.eqBox.append(self.li.pop())
            self.eqSort()
        else:
            self.mBox.append(self.li.pop())

        for sText in self.mBox:
            print(sText[0])
```

li : 문자열 길이 를 포함한 Dictionary
mBox : Main list sorting된 결과값 반환
eqBox : 문자열 길이가 같은 경우 sorting을 위한 공간

Li dictionary 길이로 정렬

정렬된 list로 같은 길이를 가진 원소 탐색

같은 길이를 가진 원소 eqBox로 저장 and 정렬

정렬된 eqBox값을 mBox 원소로 저장

No.5

문제 2

5개의 좌표와 기준좌표의 순번을 입력 받아서 기준 좌표에서 거리가 짧은 순서대로 정렬하여 출력하세요.(좌표 값은 정수, 좌표는 ()로 표현하고 빈문자열로 구분)

예)

입력:

(1,1) (2,2) (4,4) (5,5) 2

No.5

```
import re
import numpy as np
#import sys

def diff(li):

    sta = int(li.pop())-1
    convert = lambda x : int(x)
    Arr = []

    for i in li:
        Arr.append(list(map(convert,re.findall('\d', i))))

    npArr = np.array(Arr)
    value = Arr[sta]
    dist = np.sqrt((pow(npArr - value,2).sum(axis = 1)))
    return(npArr[dist.argsort()])

if __name__ == "__main__":

    li = input("좌표입력 : ").split(' ')
    for i in diff(li):
        print(i)
```

Li : 좌표

Sta : 기준 값 요소번호

Arr : 문자열로 입력 된값을 int로 변환

npArr : numpy 배열로 변환

유클리드 거리 공식

$$\sqrt{(Ax - Bx)^2 + (Ay - By)^2}$$

거리공식 적용 후

numpy의 argsort메소드로 요소 번호 추출

가까운 순으로 return