

알고리즘 응용

- week 8 -

제출일자	2021.04.28
분 반	00
이 름	강인한
학 번	201701969

제가 해결한 방법은 한 숫자씩 추가하고 위치를 바꿔가며 비교하는 보통의 방법과 다르게 작성했다는 것을 먼저 말씀드려야 할 것 같습니다. 저의 문제 해결 방법은 다음과 같습니다.

123과 45 두 숫자가 입력되었을 때, 가장 큰 숫자를 만드는 방법은

1. 숫자를 반복해 모든 숫자의 길이를 같게 맞춰 줍니다.

123 -> 123123, 45->454545

2. 같아진 숫자들을 sorted 와 reverer 함수를 통해 내림차순으로 정렬합니다.

[454545,123123]

3. 숫자들을 다시 기존의 숫자로 바꿔줍니다.

[45, 123]

4. 숫자를 연결해줍니다.

result = 45123

```
C:\Users\user\PycharmProjects\pythonProject\venv\Script
5
12 34 56 7 8
87563412
Process finished with exit code 0
```

```
def sort_desc(lst): #내림차순으로 정렬하기 위한 코드
    copy_lst = lst[:]
    for i in range(len(copy_lst)): #숫자의 길이를 최대 길이로 모두 맞춰준다.
        maxLength = max(array2)
        multiValue = int(maxLength / array2[i])
        copy_lst[i][0] = str(copy_lst[i][0]) * multiValue

    copy_lst = sorted(copy_lst) #길이가 맞춰준 숫자를 크기별로 정렬한다.

    for i in range(len(copy_lst)): #일정했던 숫자들을 다시 원래 상태로 되돌린다.
        copy_lst[i][0] = copy_lst[i][0][:copy_lst[i][1]]

    copy_lst.reverse() #내림차순으로 하기 위해 reverse해준다.

    return copy_lst
```

sort_desc 함수는 숫자를 내림차순으로 정렬하기 위한 코드로 문제 해결방법에서 설명했던 4가지 단계 중 1,2,3번을 포함하고 있습니다. 숫자의 길이를 최대 길이로 맞추고 자릿수가 같아진 모든 숫자를 sorted 함수를 통해 정렬합니다. 그 후 기존의 상태로 바꿔줍니다.

```
def paste_number(array): #내림차순으로 정렬된 숫자들을 연결해주는 작업
    result = ''
    for i in range(len(array)):
        result = result + array[i][0]
    return result
```

문제 해결 방법 4가지 단계 중 4번째에 해당하는 함수입니다. 단지 숫자를 이어주기만 하면 됩니다.

```
array_desc=sort_desc(array3) #숫자들을 합쳤을때 제일 크게되도록 정렬
rult= paste_number(array_desc) #정렬된 숫자들을 연결
print(int(rult))
```

숫자를 합쳤다고 가정할 때 가장 큰 값이 나올 수 있도록 정렬해주는 함수인
sort_desc함수를 실행하고 정렬된 숫자들을 연결해주는 동작을 하는
paste_number함수를 이용해 숫자들을 연결합니다.

4	[시간 복잡도]
---	----------

입력된 숫자의 개수를 n 이라고 했을 때 sort_desc 함수에서는 1중 for 문을 3개
쓰므로 $3*n$ 의 시간 복잡도를 갖습니다. 즉 $O(n)$ 의 시간 복잡도를 갖고
paste_number 함수에서도 for 문을 한 개만 사용하므로 $O(n)$ 의 시간 복잡도를
갖습니다. 숫자를 입력받는 부분에서도 $O(n)$ 의 시간 복잡도를 가지므로 코드
전체의 시간 복잡도는 $O(n)$ 입니다.