

24일 (목)

1. 프로그래머스 1단계 문제풀이

1) 두 개 뽑아서 더하기 (못 푼 문제) - 조합 구현 문제

두 개 뽑아서 더하기

문제 설명

정수 배열 `numbers`가 주어집니다. `numbers`에서 서로 다른 인덱스에 있는 두 개의 수를 뽑아 더해서 만들 수 있는 모든 수를 배열에 오름차순으로 담아 `return` 하도록 `solution` 함수를 완성해주세요.

제한사항

- `numbers`의 길이는 2 이상 100 이하입니다.
 - `numbers`의 모든 수는 0 이상 100 이하입니다.

입출력 예

numbers	result
[2,1,3,4,1]	[2,3,4,5,6,7]
[5,0,2,7]	[2,5,7,9,12]

입출력 예 설명

입출력 예 #1

- $2 = 1 + 1$ 입니다. (1이 `numbers`에 두 개 있습니다.)
- $3 = 2 + 1$ 입니다.
- $4 = 1 + 3$ 입니다.
- $5 = 1 + 4 = 2 + 3$ 입니다.
- $6 = 2 + 4$ 입니다.
- $7 = 3 + 4$ 입니다.
- 따라서 `[2,3,4,5,6,7]` 을 `return` 해야 합니다.

인터넷 참고 답안

```
def solution(numbers):
    answer = []

    result = []    # TIL result = list() 로도 쓸 수 있다.

    for i in range(len(numbers)):
```

```

        for j in range(i+1, len(numbers)): # TIL 'i+1' 부분은 생각하지 못했던 부분이다.
            if numbers[i] + numbers[j] not in result:
                result.append(numbers[i] + numbers[j])

    answer = sorted(result)

    return answer

```

```

# 다른사람 풀이

def solution(numbers):
    answer = []
    for i in range(len(numbers)):
        for j in range(i+1, len(numbers)):
            answer.append(numbers[i] + numbers[j])
    return sorted(list(set(answer)))

```

```

#콤비네이션 메서드 사용

from itertools import combinations

def solution(numbers):
    answer = set()
    for i in list(combinations(numbers,2)):
        answer.add(sum(i)) # TIL set 은 append 가 아니라 add 를 이용하여 세트 내부에 원소를 추가한다.
    return sorted(answer) #TIL sorted 내부에 집합도 들어갈 수 있으며 , 리스트로 변경하여 반환한다.

```

— 배운 점 —

sorted(list(set(answer))) 와 같이 리스트를 세트로, 세트를 리스트로 서로 바꿀 수 있다.

itertools 라이브러리 와 combinations 함수 → combinations(조합 배열, 조합할 개수) 위와 같이 for 문과 같이 쓰면 될 것 같다.

TIL set 은 append 가 아니라 add 를 이용하여 세트 내부에 원소를 추가한다.

#TIL sorted 내부에 집합도 들어갈 수 있으며 , 리스트로 변경하여 반환한다.

2) 소수 만들기 (2시간 걸린 문제)

소수 만들기

문제 설명

주어진 숫자 중 3개의 수를 더했을 때 소수가 되는 경우의 개수를 구하려고 합니다. 숫자들이 들어있는 배열 nums가 매개변수로 주어질 때, nums에 있는 숫자들 중 서로 다른 3개를 골라 더했을 때 소수가 되는 경우의 개수를 return 하도록 solution 함수를 완성해주세요.

제한사항

- nums에 들어있는 숫자의 개수는 3개 이상 50개 이하입니다.
- nums의 각 원소는 1 이상 1,000 이하의 자연수이며, 중복된 숫자가 들어있지 않습니다.

입출력 예

nums	result
[1,2,3,4]	1
[1,2,7,6,4]	4

입출력 예 설명

입출력 예 #1

[1,2,4]를 이용해서 7을 만들 수 있습니다.

입출력 예 #2

[1,2,4]를 이용해서 7을 만들 수 있습니다.

[1,4,6]을 이용해서 11을 만들 수 있습니다.

[2,4,7]을 이용해서 13을 만들 수 있습니다.

[4,6,7]을 이용해서 17을 만들 수 있습니다.

```
from itertools import combinations

def solution(nums):
```

```

# 조합하여 더한 숫자 구하기
combi_ls = list()

for i in list(combinations(nums, 3)):
    combi_ls.append(sum(i))

# 소수 여부 판단

for i in combi_ls[:]:          # TIL for 문에서 list remove를 할 땐, 리스트를 [:]로 복사하여 사용해야 된다.
    for j in range(2, i//2):
        if i % j == 0:        # TIL 한시간 반동안 헤 맨 부분...
            combi_ls.remove(i)
            break

return len(combi_ls)

```

```

# 다른 사람 풀이

def solution(nums):

    from itertools import combinations as cb # TIL from import 함수 안에도 쓸 수 있구나
    answer = 0                               # as 를 이용하여 간략하게 별칭을 지정할 수 있다.
    for a in cb(nums, 3):                    # cb = combinations( )
        cand = sum(a)
        for j in range(2, cand):
            if cand%j==0:
                break
        else:
            answer += 1
    return answer

```