

알고리즘 스터디 풀이 - 2회차

권민준

9020번 - 골드바흐의 추측

[에라토스테네스의 체](#)를 알고있으면 매우매우 간단하게 짤 수 있습니다. 자세한 설명은 위키백과 보시면 됩니다.

N의 숫자가 들어왔을 때 두 소수의 차이가 적은 2개의 합으로 표현이 된다는 것은 $N/2$ 에 가장 가까운 두개의 소수를 찾으면 됩니다. 이를 가지고 표현해보면 다음과 같아요

1. $N/2$ 에서 1씩 줄여가면서 소수를 찾습니다.
2. $N/2$ 보다 작은 소수를 K라고 하면 $N - K$ 가 소수이면 그게 두 소수의 차이가 소수들입니다.

이렇게 하면 끝입니다 :) 다음은 파이썬 코드입니다.

```
T = int(input())
E = [True] * 10001

# Build 에라토스테네스의 체
# n의 최대 약수가 sqrt(n) 이하이므로 i=sqrt(n)까지 검사
m = int(10000 ** 0.5)
for i in range(2, m + 1):
    if E[i] == True: # i가 소수인 경우
        for j in range(i+i, 10001, i): # i이후 i의 배수들을 False 판정
            E[j] = False

for _ in range(T):
    N = int(input())
    for k in range(N//2, 1, -1):
        if E[k] and E[N-k]:
            print(k, N-k)
            break
```

5568번 - 카드 놓기

N이 최대 10에 K가 최대 4 밖에 안되므로 순열을 모두 따져도 $O(10^4)$ 면 모든 계산이 끝납니다. 즉, 제한시간 1초내에 모든걸 끝낼 수 있다는 것을 알 수 있습니다.

만들어진 내용은 최대 5040개인데 여기서 중복을 제외하면 값을 바로 낼 수 있습니다.

```
from itertools import permutations
N = int(input())
K = int(input())
L = []
result = []
for _ in range(N):
    L.append(input())
for x in permutations(L, K):
    r = ''
    for i in range(K):
        r += x[i]
    result.append(r)
print(len(set(result))) # set은 값의 Hash로 구분하기 때문에 중복을 허용하지 않습니다. 이를
# 통해 중복제거를 하면 쉽게 풀립니다. 유용한 Trick & Tip 이에요.
```

프로그래머스 42840번 - 모의고사

3명의 수포자가 찍는 규칙을 주고 누가 가장 많이 맞는지를 배열로 감싸서 return 하는 문제입니다. 명세가 너무 뚜렷하고 그냥 구현의 문제라서 따로 풀이라고 쓸만한게 없네요.. 수포자가 3명이라는 게 정해져있고, 찍는 규칙도 상수로 주어져서 제가 봤을 때 별로 좋은 문제는 아닌것 같습니다.

```
def solution(answers):
    L = [[1, 2, 3, 4, 5], [2, 1, 2, 3, 2, 4, 2, 5], [3, 3, 1, 1, 2, 2, 4, 4, 5, 5]]
    R = [[1, 0], [2, 0], [3, 0]]
    for i in range(len(answers)):
        answer = answers[i]
        for j in range(3):
            if L[j][i%len(L[j])] == answer:
                R[j][1] += 1
    R = sorted(R, key = lambda x: x[1], reverse = True)
    result = []
    M = max(R, key=lambda x: x[1])[1]
    for x in R:
        if x[1] == M:
            result.append(x[0])
    return result
```

프로그래머스 42842번 - 카펫

카펫에서 빨간색이 존재하기 위해서는 가로 혹은 세로 길이가 무조건 3이상이어야 존재합니다. 이를 통해서 3부터 가로 길이를 키워보면서 직접 빨간색의 개수를 센 뒤 맞는 것이 있는지 확인합니다.

```
def solution(brown, red):
    L = brown + red
    for w in range(3, L//2+1):
        if w*(L//w) == L and L-(w*2+(L//w)*2-4) == red:
            return [L//w, w]
```