알고리즘 Week 5

17기 정규세션

TOBIG’S 16기 김권호

Contents

Unit 01 | 3주차 과제 리뷰

Unit 01 | 3주차 과제 리뷰

17기 정규세션
TOBIG'S 16기 김권호

문제 1. 숫자 카드

각 카드에는 양의 정수가 쓰여 있다. N장의 카드가 주어졌을 때, M을 넘지 않으면서 M에 최대한 가까운 카드 3장의 합을 구해 출력하시오.

[입력 조건]
- 첫째 줄에 카드의 개수 N과 M이 주어진다. (단, 3 ≤ N ≤ 100 이며, 10 ≤ M ≤ 300,000 이다.)
- 둘째 줄에는 카드에 쓰여 있는 수가 주어진다.
- 합이 M을 넘지 않는 카드 3장을 찾을 수 있는 경우만 입력으로 주어진다.

[출력 조건]
- 첫째 줄에 M을 넘지 않으면서 M에 최대한 가까운 카드 3장의 합을 출력한다.

[입력 예시]
10 500
93 181 245 214 315 36 185 138 216 295

[출력 예시]
497

Unit 01 | 3주차 과제 리뷰

17기 정규세션
TOBIG'S 16기 김권호

문제 1. 숫자 카드

- 카드의 개수 N개를 모두 3개씩 조합
- 합을 구하고 내림차순 정렬
- 높은 수부터 M과 비교하며 M보다 크면 pass
- M보다 card_sum_lst의 값이 작아지는 순간 return

```python
from itertools import combinations

N, M = map(int, input().split())
card_lst = list(map(int, input().split()))

card_tup_lst = list(combinations(card_lst, 3))
card_sum_lst = []
for tup in card_tup_lst:
    card_sum_lst.append(sum(tup))
card_sum_lst.sort(reverse = True)

for val in card_sum_lst:
    if val > M:
        continue
    else:
        result = val
        break

print(result)
```

17기 정규세션
TOBIG'S 16기 김권호

문제 2. 방탈출 게임

승주는 방탈출 게임을 하고 있다. 승주가 서랍에서 한자리 숫자가 적힌 종이 조각들을 발견했다.
흩어진 종이 조각을 붙여 소수를 몇 개 만들 수 있는지 알아내면 탈출을 위한 열쇠를 획득할 수 있다.
각 종이 조각에 적힌 숫자들로 이루어진 문자열이 주어졌을 때, 종이 조각으로 만들 수 있는 소수가 몇 개인지 출력하시오.

[입력 조건]
- 첫째 줄에 각 종이 조각에 적힌 숫자들로 이루어진 문자열이 주어진다.
  (단, 문자열의 길이는 1 이상 7 이하이고, 종이 조각에 적힌 각 숫자는 0~9로 이루어져 있다.)
- "025"은 0, 2, 5 숫자가 적힌 종이 조각이 흩어져있다는 의미이다.

[출력 조건]
- 첫째 줄에 종이 조각으로 만들 수 있는 소수의 개수를 출력한다.

[입력 예시1]
"011"

[출력 예시1]
2

[입력 예시2]
"17"

[출력 예시2]
3

Unit 01 | 3주차 과제 리뷰

17기 정규세션
TOBIG'S 16기 김권호

문제 2. 방탈출 게임

- 주어진 숫자들로 만들 수 있는 모든 숫자 조합을 순열 조합을 통해 구한다.
- 만들어진 순열 조합 list를 순회하면서 소수 판별 검사를 진행한다.

```
import math
        int(math.sqrt(prime))
```

- 특정한 숫자의 제곱근 까지만 약수의 여부를 검증하면 O(N^1/2)의 시간 복잡도로 소수 판별이 가능하다.
- 8의 경우 약수는 1,2,4,8인데, 2와 4는 대칭이므로 2까지만 검사해도 소수 판별이 가능하다.

중복제거

```python
from itertools import permutations

def solution(numbers):
    num = list(numbers); perm = []
    for i in range(1,len(num)+1):
        perm+=permutations(num, i)
    n = [int(''.join(perm[i])) for i in range(len(perm))]

    count = 0
    flag = True
    for prime in set(n):
        if prime==2: count+=1
        elif prime>2:
            for i in range(2, prime):
                if prime%i!=0: continue
                else: flag = False; break;
            if flag == True:count+=1
        flag = True
    return count

numbers = input()
print(solution(numbers))
```

17기 정규세션
TOBIG'S 16기 김권호

문제 3. 티셔츠 갈아입기

투빅스에서 전 기수 엠티를 가서 대강당에서 게임을 한다. 총 M*N명의 투빅이들은 빨간 티셔츠 아니면 파란 티셔츠를 입고 있다. M × N 행렬의 형태로 M*N개의 의자가 놓여져 있는 강당에서 투빅이들은 무작위로 앉아 있다.

[게임 룰]

1. 64명의 투빅이들은 8 × 8 정사각형 형태로, 서로 다른 색깔의 티셔츠를 입고 번갈아 앉아있어야 한다.
   (이때, 8 × 8 정사각형의 위치에는 제약이 없다. M × N 행렬 내에만 들어가있으면 된다.)

2. 8 × 8 정사각형 형태의 각 64개의 의자에 앉아 있는 투빅이들은 빨간 티셔츠나 파란 티셔츠 중 하나를 입고 있어야 하고, 이웃하는 투빅이끼리는 서로 다른 색깔의 옷을 입고 있어야 한다.

위 게임의 룰을 참고하여, 64명의 투빅이들이 알맞게 티셔츠를 입고 8 x 8 정사각형 형태로 앉아 있기 위해 티셔츠를 갈아입어야 하는 최소 횟수(명수)를 출력하시오.

Unit 01 | 3주차 과제 리뷰

17기 정규세션
TOBIG'S 16기 김권호

[입력 조건]
- 첫째 줄에 M과 N이 주어진다. M과 N은 8보다 크거나 같고, 50보다 작거나 같은 자연수이다.
- 둘째 줄부터 M × N 형태로 의자에 앉은 투빅이들의 옷 색깔 상태가 주어진다. R은 빨간색이며, B는 파란색이다.

[출력 조건]
- 첫째 줄에, 64명의 투빅이들이 알맞게 티셔츠를 입고 8 x 8 정사각형 형태로 앉아 있기 위해 티셔츠를 갈아입어야 하는 최소 횟수(명수)를 출력하시오.

[입력 예시]

```
10 13
RRRRRRRRBRBRB
RRRRRRRRRBRBR
RRRRRRRRBRBRB
RRRRRRRRRBRBR
RRRRRRRRBRBRB
RRRRRRRRRBRBR
RRRRRRRRBRBRB
RRRRRRRRRBRBR
BBBBBBBBBBRBR
BBBBBBBBBBRBR
```

[출력 예시]

```
12
```

R	B	R	B	R	B	B	B	R	R
R	B	R	R	B	R	R	B	R	R
B	R	B	B	R	R	R	R	R	R
R	R	B	R	R	R	B	B	R	R
R	R	R	R	B	R	R	R	B	B
R	R	R	B	R	R	R	R	R	R
R	R	R	R	B	R	R	R	B	R
B	R	R	R	R	R	R	R	R	R
R	R	B	R	R	R	B	R	B	R

Unit 01 | 3주차 과제 리뷰

17기 정규세션
TOBIG'S 16기 김권호

문제 3. 티셔츠 갈아입기

- 주어진 티셔츠 배열을 8x8범위로 순회한다.
- R로 시작하는 경우(idx1)와 B로 시작하는 경우(idx2) 갈아입어야 할 사람 수를 모두 기록하여 배열(mini)에 저장한다.
- mini -> 8x8 범위마다, R로 시작하는 경우와 B로 시작하는 경우 각각에 대해 갈아입어야 하는 사람 수가 저장되어 있다.
- 모든 순회가 끝난 후, 저장된 배열에서 최솟값을 출력

```python
M, N = map(int, input().split())
l = []
mini = []

for _ in range(M):
    l.append(input())

for a in range(M - 7):
    for i in range(N - 7):
        idx1 = 0
        idx2 = 0
        for b in range(a, a + 8):
            for j in range(i, i + 8):              # 8X8 범위를 B와 R로 번갈아가면서 검사
                if (j + b)%2 == 0:
                    if l[b][j] != 'R': idx1 += 1
                    if l[b][j] != 'B': idx2 += 1
                else :
                    if l[b][j] != 'B': idx1 += 1
                    if l[b][j] != 'R': idx2 += 1
        mini.append(idx1)                          # R로 시작했을 때 칠해야 할 부분
        mini.append(idx2)                          # B로 시작했을 때 칠해야 할 부분

print(min(mini))                                   # 칠해야 하는 개수의 최소값
```

17기 정규세션
TOBIG'S 16기 김권호

Unit 02 | 동적계획법

Unit 02 | 동적계획법

17기 정규세션
TOBIG'S 16기 김권호

1주차	OT & 알고리즘 기초
3주차	완전탐색
5주차	동적계획
7주차	분할정복
9주차	탐욕 알고리즘

17기 정규세션
TOBIG'S 16기 김권호

동적계획법 (dynamic programming)

- 입력 크기가 작은 부분 문제들을 해결한 후, 해당 부분 문제의 해를 활용해서, 큰 부분 문제를 해결하고 최종적으로 전체 문제를 해결하는 알고리즘 기법
- 상향식 접근과 하향식 접근 두 가지 방법이 존재
- 상향식 접근법으로 가장 최하위 해답을 구한 후 이를 저장하고, 해당 결괏값을 이용해 상위 문제를 풀어가는 방식을 활용
- Memoization 기법 활용(문제를 잘게 쪼갤 때, 부분 문제는 중복되어 재활용).

17기 정규세션
TOBIG'S 16기 김권호

동적계획법 (dynamic programming)

동적 프로그래밍 개념을 문제에 적용하는 과정

- 부분 문제 정의하기
  - 점화식 만들기, 부분 문제의 개수 세기, 각 문제의 선택지 개수 세기(예외처리, 탈출조건)
- 메모 테이블 구성하기
- 부분 문제의 선후관계(topological order)를 고려해 순서대로 실행하기

하향식 vs 상향식

- 하향식: 현재 문제를 처리하기 위해 부분 문제로 나누어 재귀적으로 내려가는 방식
- 상향식: 위상적 순서에 따라 부분 문제들을 해결해 올라오는 방식
- (오름차순/내림차순의 차이가 아니라, 부분 문제들을 푸는 순서의 차이)

재귀 함수의 경우 메모리 문제가 발생할 수 있으므로 상향식 방법의 사용을 많이 사용한다

17기 정규세션
TOBIG'S 16기 김권호

재귀 (Recursion)

함수 안에 자신의 함수를 다시 호출하는 함수를 의미.

재귀함수는 자신의 로직을 내부적으로 반복하다가, 일정한 조건이 만족되면 함수를 이탈하여 결과를 도출한다

```
def factorial(x):
    if x > 1:
        return x * factorial(x - 1)

    else:
        return x
```

factorial(3) = 3* factorial(2)

factorial(2) = 1 * factorial(1)

factorial(1) =1

Unit 02 | 동적계획법

17기 정규세션
TOBIG'S 16기 김권호

EX) Fibonacci numbers

예외처리

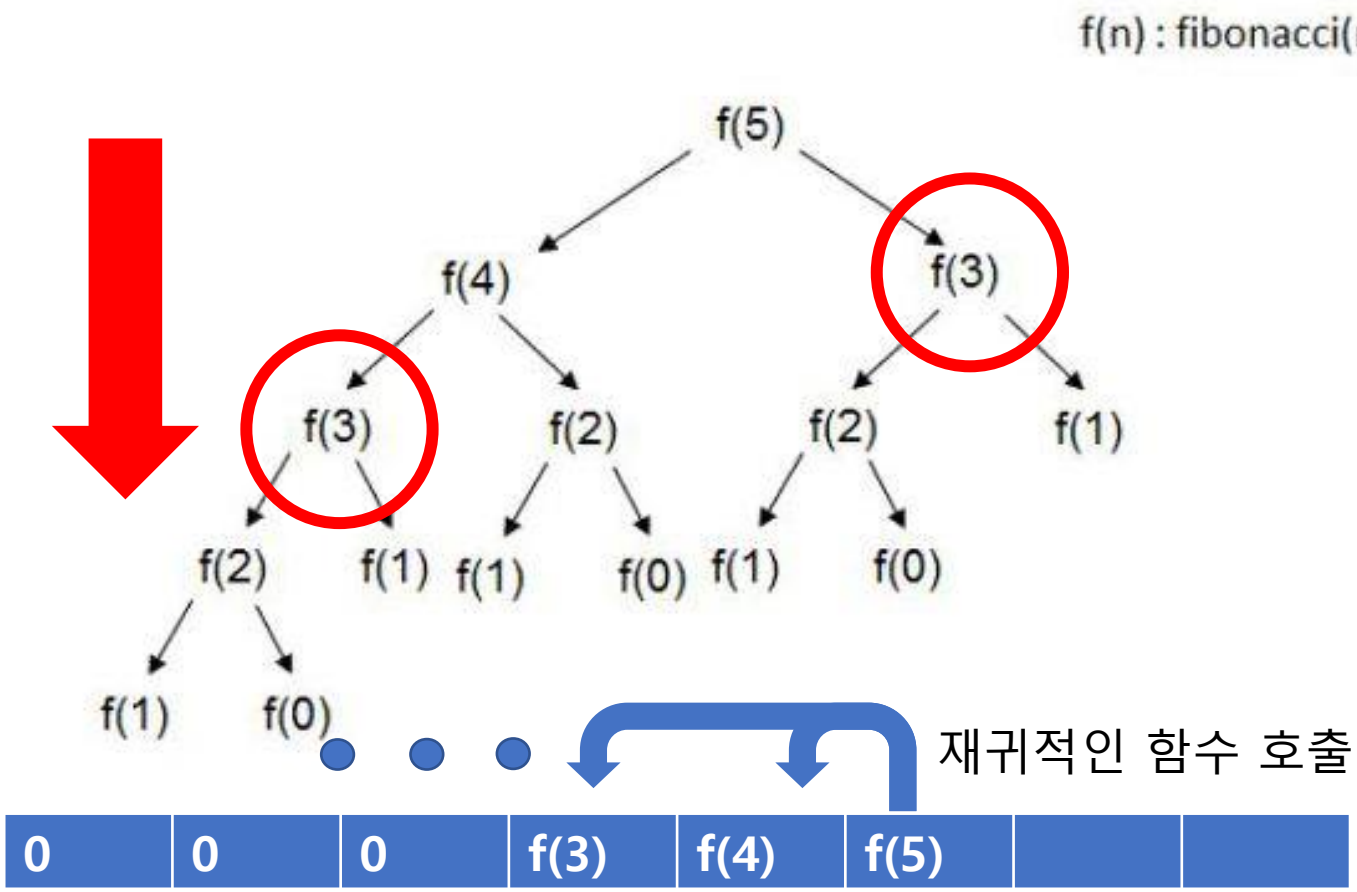
$$f(n) = \begin{cases} 0 & if \quad n = 0 \\ 1 & if \quad n = 1 \\ F(n-1) + F(n-2) & if \quad n > 1 \end{cases}$$

점화식

2개의 부분문제

Unit 02 | 동적계획법

EX) Fibonacci numbers(하향식)



dp[i] = dp[i-1] + dp[i-2]

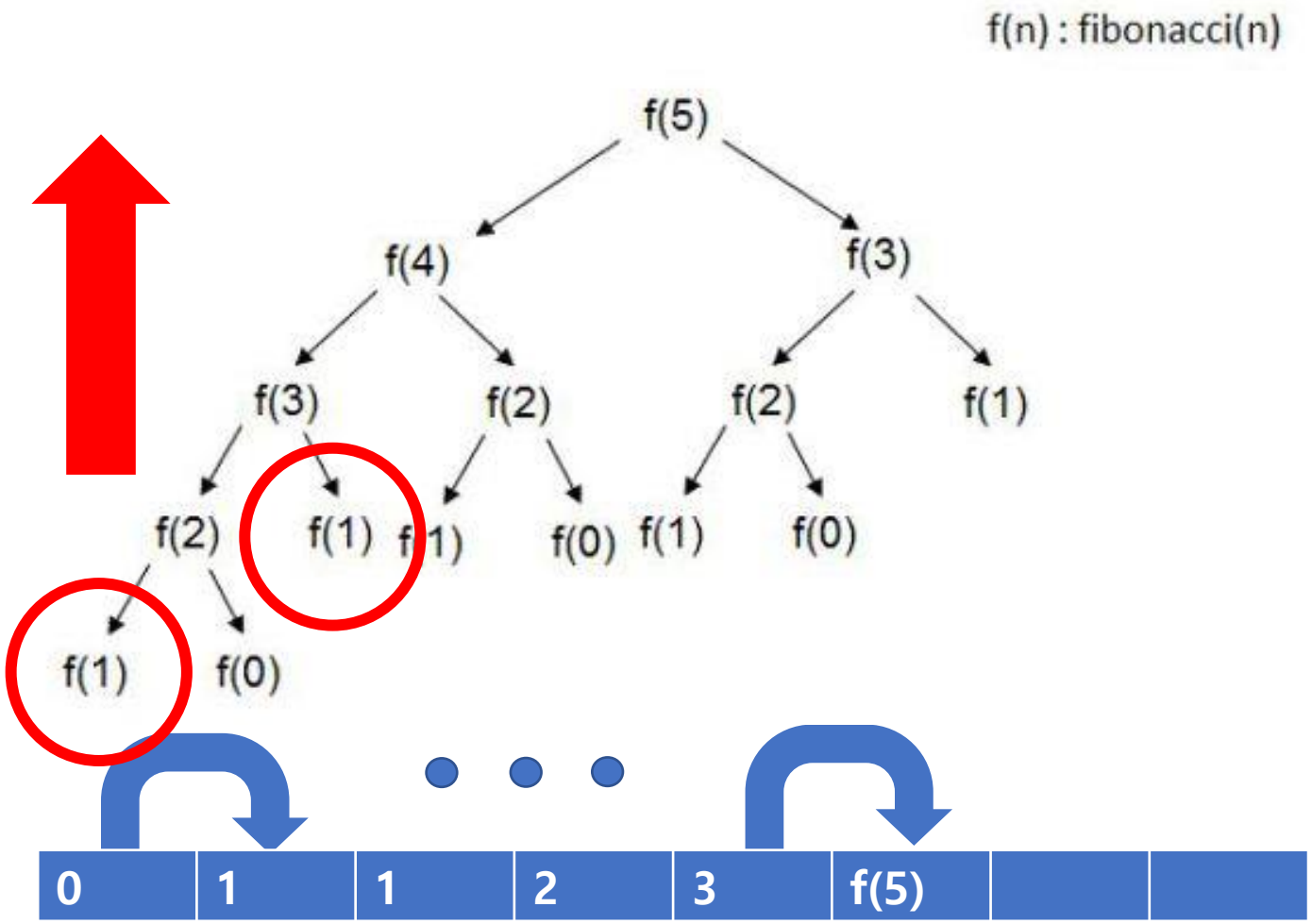
f(5) = f(4) + f(3)
f(4) = f(3) + f(2)
f(3) = f(2) + f(1)

```
Dp = [0 for i in range(n+1)]
def fibo(Dp, n):
    if Dp[n] != 0:
        return Dp[n]
    elif n == 1 or n == 2:
        Dp[n] = 1
    else:
        Dp[n] = fibo(Dp, n-1) + fibo(Dp, n-2)
    return Dp[n]
```

Unit 02 | 동적계획법

17기 정규세션
TOBIG'S 16기 김권호

EX) Fibonacci numbers(상향식)



$$dp[i] = dp[i-1] + dp[i-2]$$

f(0) = 0
f(1) = 1
f(2) = 1
f(3) = f(1) +f(2)

```
def fibo(n):
    Dp = [0, 1, 1]
    for i in range(3, n + 1):
        Dp.append(Dp[i-1] + Dp(i-2))
    return Dp[n]
```

Unit 03 | 5주차 과제 소개

17기 정규세션
TOBIG'S 16기 김권호

문제 1. 쇼핑몰과 박스

투빅이는 쇼핑몰을 운영하고 있다. 매장에 옷을 포장하기 위한 박스로는 1kg, 2kg, 3kg 용량의 박스가 있다. 현재 포장해야하는 옷의 무게가 nkg라고 할 때, 주어진 박스를 사용하여 옷을 포장하는 방법의 수를 구하시오. 단, 주어진 무게의 옷은 모두 박스 안에 포장되어야 한다.

[입력 조건]
- 첫째 줄에 옷을 배송해야 할 주소의 개수가 주어진다. 주소마다 배송해야 할 옷 무게는 각각 한 줄로 이루어져 있고, 옷의 무게 n이 주어진다. n은 양수이며 11보다 작다.

[출력 조건]
- 포장해야하는 옷 마다, 적절히 박스를 사용해 포장하는 방법의 수를 출력한다.

[입력 예시]
3
4 7 10

[출력 예시]
7 44 274

4-> 1+1+1+1, 1+1+2, 1+2+1, 2+1+1, 2+2, 1+3, 3+1 (7가지)

17기 정규세션
TOBIG'S 16기 김권호

문제 2. 복세편살(복잡한 세상 편하게 살자!)

투빅스 회사에는 처리해야 할 문서가 N장이 있다. 회사에는 사원이 3명 있는데, 각각 한나, 예림, 승주이다.
한나는 N이 3의 배수일 경우, 해당 문서의 1/3만 남기고 모두 처리한다.
예림이는 N이 2의 배수일 경우, 해당 문서의 1/2만 남기고 모두 처리한다.
승주는 남은 문서의 개수에 관계 없이 문서를 1장 처리한다.
이 때, N장의 문서 중 마지막 문서는 사장의 결제가 필요해 그대로 사장인 권호에게 보여주려고 한다.
1장을 남기고 나머지 문서를 모두 처리하기 위해 한나, 예림, 승주가 일하는 최소한의 횟수를 구하시오

[입력 조건]
- 1≤N≤10^6인 문서 개수 N이 주어진다.

[출력 조건]
- 첫째 줄에 사원을 최소한으로 일하게 하는 방법의 수를 출력한다.

[입력 예시] 10 [출력 예시] 3

10-> 10-1=9(승주) -> 9/3 = 3(한나)-> 3/3(한나) =1 (3번)

17기 정규세션
TOBIG'S 16기 김권호

문제 3. 신기한 초밥집

민진이는 신기한 초밥집에 점심을 먹으러 갔다. 신기한 초밥집엔 초밥을 주문하는 특별한 규칙이 있다.

규칙: 다양한 초밥이 올려져 있는 초밥 접시가 일렬로 놓여 있다. 초밥 접시를 선택하면 접시에 놓여 있는 초밥들은 모두 먹어야 하고, 먹은 후에 접시는 원래 위치에 다시 놓아야 한다. 또한, 연속으로 놓여 있는 접시 3개를 고를 수는 없다.

배가 고픈 민진이는 될 수 있는 대로 많은 양의 초밥을 먹기 위해 어떤 접시를 선택해야 할까? 1부터 n까지의 번호가 붙어 있는 n개의 접시가 순서대로 놓여 있고, 각 접시에 놓여 있는 초밥의 개수가 주어졌을 때, 민진이를 도와 가장 많은 초밥을 먹을 수 있도록 프로그램을 작성해보자!

예를 들어 6개의 초밥 접시가 있고, 각각의 접시에 순서대로 6개, 10개, 13개, 9개, 8개, 1개 만큼의 초밥이 놓여져 있을 때, 첫 번째, 두 번째, 네 번째, 다섯 번째 접시를 선택하면 총 33개의 초밥을 최대로 먹을 수 있다.

Unit 03 | 5주차 과제 소개

17기 정규세션
TOBIG'S 16기 김권호

[입력 조건]
- 첫째 줄에 접시의 개수 n이 주어진다. (1 ≤ n ≤ 10,000)
- 둘째 줄부터 n+1번째 줄까지 접시에 놓여 있는 초밥의 개수가 순서대로 주어진다. 초밥의 개수는 1,000 이하의 음이 아닌 정수이다.

[출력 조건]
- 첫째 줄에 최대로 먹을 수 있는 초밥 개수를 출력한다.

[입력 예시]
6
6
10
13
9
8
1

[출력 예시]
33

6 10 13 9 8 1 -> 6+10+9+8 = 33
세번째 접시인 13은 앞의 두 접시를 연속으로 먹으면 먹을 수 없다!

Unit 04 | 3주차 과제 소개

17기 정규세션
TOBIG'S 16기 김권호

Course Code

6094c0d866

Course Link

https://class.mimir.io/courses/6094c0d866/registrations/new

질문 / 힌트 / 오류 등 모든 문의는
누구에게? 김권호 / 박한나 / 이승주 / 이예림 / 전민진 혹은 멘토에게!
언제? 24시간 OK

17기 정규세션

TOBIG’S 16기 김권호

알고리즘 파이팅♡