



알고리즘 Week 3

19기 정규세션

TOBIG'S 18기 이기정

Contents



17기 정규세션
TOBIG'S 16기 이예림

Unit 01 | 1주차 과제 리뷰

Unit 02 | 시간 복잡도

Unit 03 | 완전탐색

Unit 04 | 3주차 과제 소개



17기 정규세션
TOBIG'S 16기 이예림

Unit 01 | 1주차 과제 리뷰



문제 1. 승리 카드

카드들이 $N \times M$ 형태로 놓여 있을 때, 게임의 룰에 맞게 승리한 카드의 숫자를 출력하는 프로그램을 작성하시오.

[게임 룰]

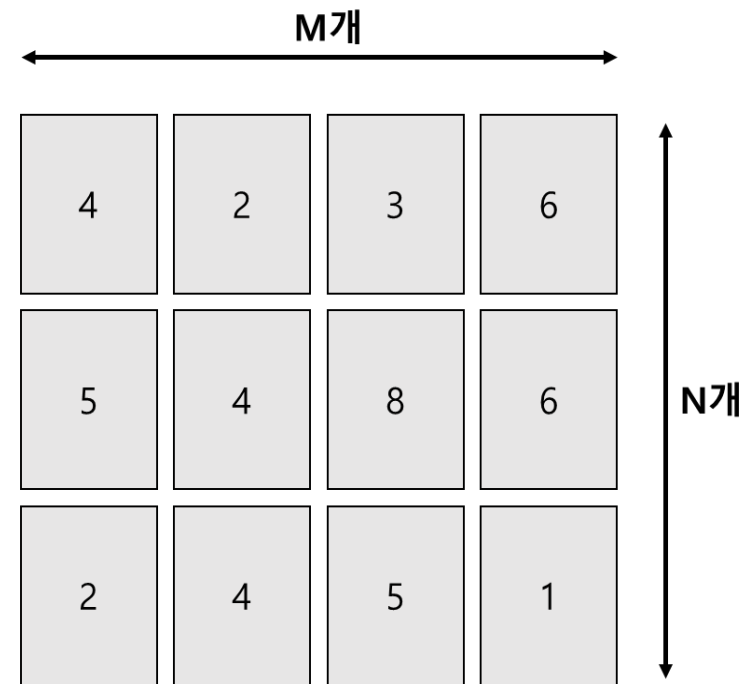
1. N 명의 투빅이들은 각각 M 개의 숫자 카드를 골라 차례대로 한 줄 씩 일렬로 놓을 수 있다. 최종적으로 $N \times M$ 형태로 숫자카드들이 놓여진다. 여기서 N 은 행의 개수를 의미하고, M 은 열의 개수를 의미한다.
2. 먼저, 자신이 놓은 카드 M 개 중 가장 작은 숫자 카드를 뽑는다.
3. 그 다음, N 명의 투빅이들이 고른 가장 작은 숫자 카드 N 개 중 가장 높은 숫자를 가진 투빅이가 승리한다.

[입력 조건]

- 첫째 줄에 숫자 카드들이 놓인 행의 개수 N 과 열의 개수 M 이 공백을 기준으로 하여 각각 자연수로 주어진다.
- 둘째 줄부터 N 개의 줄에 걸쳐 각 카드에 적힌 숫자가 주어진다. 각 숫자는 1 이상 10,000 이하의 자연수이다.

[출력 조건]

- 첫째 줄에 게임의 룰에 맞게 승리한 카드에 적힌 숫자를 출력한다.



문제 1. 승리 카드

[입력 예시]

3 4
4 2 3 6
5 4 8 6
2 4 5 1

[출력 예시]

4

```
n, m = map(int, input().split(" "))  
  
result = 0  
  
for i in range(n):  
    arr = list(map(int, input().split(" ")))  
    min_value = min(arr)  
    result = max(result, min_value)  
  
print(result)
```

↑ 문제 1 Solution Code




문제 2. 달팽이가 가자

맹구가 키우는 달팽이가 $N \times N$ 크기의 정사각형 미로 위에 서 있다. 이 공간은 1 곱하기 1 크기의 정사각형으로 나누어져 있으며 가장 왼쪽 위 좌표는 (1,1)이고, 가장 오른쪽 아래 좌표는 (N,N)이다. 맹구는 달팽이가 상, 하, 좌, 우 방향으로 스스로 이동할 수 있도록 달팽이를 훈련시키고자 한다. 달팽이의 시작 좌표는 항상 (1,1)이다. 맹구가 미로에 R, L, U, D 중의 문자를 반복적으로 적어 달팽이에게 보여주면 달팽이는 미로에 적힌 문자를 보고 차례대로 이동을 한다. 달팽이가 도착하는 최종 위치가 정답 위치라면 맹구는 달팽이에게 갯잎을 선물로 준다. 이때 달팽이가 $N \times N$ 크기의 정사각형 공간을 벗어나는 움직임은 무시된다. 그때, 주어진 입력에 따라 도착한 달팽이의 도착 지점의 좌표를 출력하세요.

다음은 $N=5$ 인 지도와 그림판 내용이다.

이 경우 6개의 명령을 받아 달팽이가 움직이는 위치는 순서대로 (1,2), (1,3), (1,3), (2,3), (3,3), (3,2) 이므로 최종적으로 달팽이가 도착하게 되는 곳의 좌표는 (3,2)이다. 다시 말해 3행 2열의 위치에 해당하므로 (3,2)라고 적는다.

(1,1) 	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

명령 : R R U D D L

- L: 왼쪽으로 한 칸 이동
- R: 오른쪽으로 한 칸 이동
- U: 위로 한 칸 이동
- D: 아래로 한 칸 이동



문제 2. 달팽이야 가자

[입력 조건]

- 첫째 줄에 공간의 크기를 나타내는 N 이 주어진다. ($1 \leq N \leq 100$)
- 둘째 줄에 달팽이가 이동해야 하는 방향이 적힌 그림판 내용이 주어진다. ($1 \leq \text{이동 횟수} \leq 100$)

[출력 조건]

- 첫째 줄에 달팽이가 최종적으로 도착해야 하는 지점의 좌표 (X, Y) 를 공백으로 구분하여 출력한다.

[입력 예시]

5
R R U D D L

[출력 예시]

3 2

```
N = int(input())
plan = input().split()
x, y = 1, 1

type = ['L', 'R', 'U', 'D']

for i in range(len(plan)):
    if plan[i] == type[0]:
        if y - 1 == 0:
            continue
        else:
            y = y - 1
    if plan[i] == type[1]:
        if y + 1 == N+1:
            continue
        else:
            y = y + 1
    if plan[i] == type[2]:
        if x - 1 == 0:
            continue
        else:
            x = x - 1
    if plan[i] == type[3]:
        if x + 1 == N+1:
            continue
        else:
            x = x + 1

print(x, y)
```

↑ 문제 2 Solution Code

문제 3. 가보자고

소영이의 대학원 건물 엘리베이터는 한 번 작동 시 U층만큼 올라갔다가, D층만큼 내려간다.
이 엘리베이터를 타고 건물 최상층인 H층까지 올라가려고 한다.
건물은 0층부터 H층까지로 구성되어 있으며, H층에 도착하면 엘리베이터가 내려가지 않는다. 소영이가 이 엘리베이터를 타고 H층에 도착하려면, 몇 번 동작 시켜야 할까요?

[입력 조건]

- 첫째 줄에 세 정수 U, D, H가 공백으로 구분되어 주어진다.
- $1 \leq D \leq U \leq H \leq 1000000000$

[출력 조건]

- 첫째 줄에 H층에 도착하기까지 엘리베이터를 작동시켜야 하는 횟수를 출력한다.

[입력 예시1]

2 1 5

[출력 예시1]

4

[입력 예시2]

5 1 6

[출력 예시2]

2

```
import math

u, d, h = map(int, input().split())
k = math.ceil((h-u)/(u-d))
print(1+k)
```

↑ 문제 3 Solution Code



문제 4. 숫자 바꾸기

재민이는 0과 1로 이루어진 카드 번호 C 를 가지고 있다. 카드 번호 C 의 숫자를 전부 같게 만들어야 하는데, 재민이가 할 수 있는 것은 카드 번호 C 에서 연속된 하나 이상의 숫자를 잡고 모두 바꾸는 것이다. 바꾸는 것은 1을 0으로, 0을 1로 뒤집는 것을 의미한다.

예를 들어 문자열 C 가 00011000일 때,

1. 전체를 바꾸면 11100111이 된다.
2. 4번째 문자부터 5번째 문자까지 바꾸면 1111111이 되어서 2번 만에 모두 같은 숫자로 만들 수 있다.

하지만, 처음부터 4번째 문자부터 5번째 문자를 바꾸면 한 번에 0000000이 되어서 1번 만에 같은 문자로 뒤집을 수 있다. 문자열이 주어졌을 때, 재민이가 해야 하는 행동의 최소 횟수를 출력 하시오.

문제 4. 숫자 바꾸기

[입력 조건]

첫째 줄에 카드 번호 C가 주어진다. 카드 번호 C의 길이는 100만보다 작다.

[출력 조건]

첫째 줄에 재민이가 해야 하는 행동의 최소 숫자를 출력한다.

[입력 예시]

11111

[출력 예시]

0

```
str = input()
count = [0] * 2
count[int(str[0])] = 1
for i in range(1, len(str)):
    if str[i] != str[i - 1]:
        count[int(str[i])] += 1
print(min(count))
```

Ex) 0/1/0/1/0/11

0 -> count[0]:1, count[1]:0

01 -> count[0]:1, count[1]:1

010 -> count[0]:2, count[1]:1

0101 -> count[0]:2, count[1]:2

01010 -> count[0]:3, count[1]:2

010101 -> count[0]:3, count[1]:3

0101011 -> count[0]:3, count[1]:3

Answer: 3



문제 5. 우영우수

영우는 자신의 이름과 같이 거꾸로 읽어도, 앞으로 읽어도 같은 수를 우영우수라고 부른다.

예를 들어 121, 12421을 우영우수라고 할 수 있다. 123, 1231은 뒤에서부터 읽으면 다르므로 우영우수가 아니다. 이 문제에서는 010과 같이 앞에 무의미한 0이 오는 경우가 없다고 하자.

[입력 조건]

입력은 여러 개의 테스트 케이스로 이루어져 있으며, 각 줄마다 0보다 크고 100000보다 작은 수가 주어진 다. 입력 마지막 줄에는 0이 주어지며, 이 줄은 문제에 포함되지 않는다.

[출력 조건]

각 줄마다 주어진 수가 우영우수면 'yes', 아니면 'no'를 출력한다.

[입력 예시]

817

707

0

[출력 예시]

no

yes

```
while(1):
    Sinput=input()

    if Sinput=='0': break

    Linput=[i for i in Sinput]
    Sinput2=''.join(Linput[::-1])

    if Sinput==Sinput2: print("yes")
    else: print("no")
```



17기 정규세션
TOBIG'S 16기 이예림

Unit 02 | 시간 복잡도

‘효율적인’ 알고리즘을 위한 고민

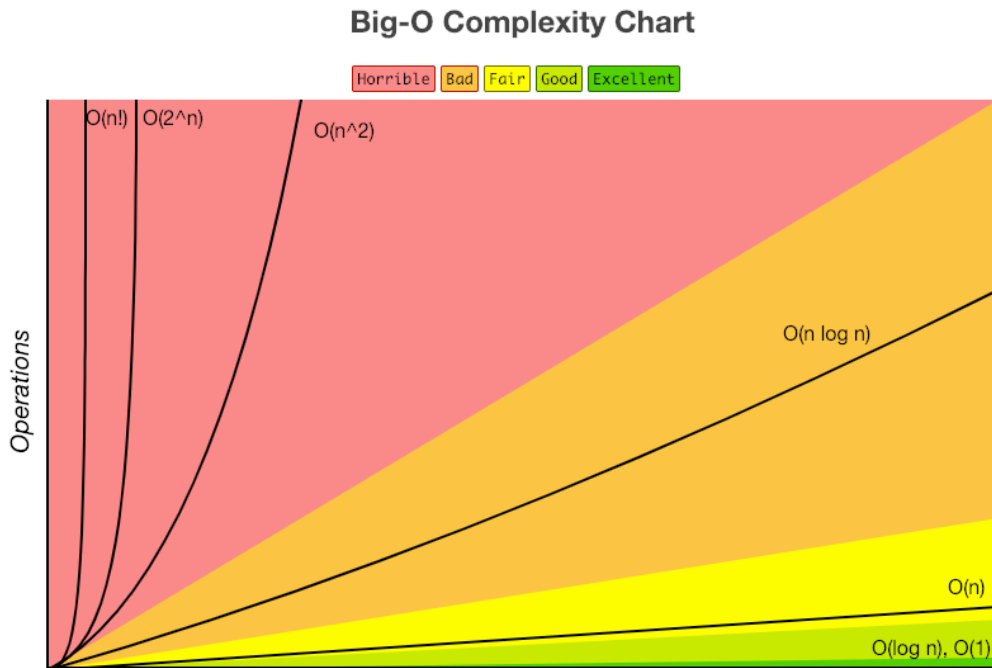
- 해답을 찾는 것도 중요하지만, ‘효율적인’ 방법으로 문제를 풀었는지도 중요
- ‘효율성’을 고려한다 = 시간 복잡도를 고려한다

시간 복잡도 (Time Complexity)

- 알고리즘을 풀어낼 때 프로세스가 수행해야 하는 연산을 수치화한 것
- 주로 Big-O 표기법을 이용해 시간 복잡도를 표현

Big-O 표기법

- 가장 영향력 있는 항을 기준으로 표기
 - 예 $n^2 + 2n + 1 \rightarrow O(n^2)$



$O(1)$

- 입력에 관계없이 동일한 시간 복잡도 (고정된 연산)
- ```
def hello_world():
 print("hello, world!")
```

### $O(n)$

- 입력값이 증가하면 처리 시간이 선형적으로 증가
- ```
def print_each(li):  
    for item in li:  
        print(item)
```

$O(n^2)$

- 반복문이 두 번 사용된 경우
- ```
def print_each_n_times(li):
 for n in li:
 for m in li:
 print(n,m)
```



17기 정규세션  
TOBIG'S 16기 이예림

## Unit 03 | 완전탐색



---

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

---



## 완전탐색 (Exhaustive Search, Brute-Force)

- 컴퓨터의 빠른 계산 능력을 이용하여 **가능한 경우의 수를 일일이 나열**하면서 답을 찾는 방법
  - 예 : 4자리 숫자로 된 암호를 찾기 위해 0000부터 9999까지 모두 입력해보기
- **종류**
  - Brute Force 기법 - 반복문 / 조건문 이용
  - 백트래킹 (Backtracking) - 가지치기, 분할정복 이용 (7주차)
  - 순열 (Permutation) - 서로 다른 N개를 일렬로 나열
  - 비트 마스크 (Bit Mask) - 이진수 이용
  - 재귀함수
  - BFS/DFS - 그래프 자료 구조에서 모든 정점을 탐색

## 완전탐색 (Exhaustive Search, Brute-Force)

- **장점**

- 직관적이므로 이해하기 쉽고, 구현도 간단하다.
- 즉, 문제의 정확한 결과값을 얻어낼 수 있는 가장 확실하며 기초적인 방법

- **단점**

- 시간 초과 문제

**BUT, 완전탐색은 효율적인 알고리즘을 짜는 근간이면서,  
코딩 테스트에서 경우의 수를 제한하는 방식 등으로 출제된다!**



17기 정규세션  
TOBIG'S 16기 이예림

## Unit 04 | 3주차 과제 소개



## 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,  $3 \leq N \leq 100$  이며,  $10 \leq M \leq 300,000$  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

## 문제 2. 방탈출 게임

기정이는 방탈출 게임을 하고 있다. 기정이가 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



## 문제 3. 티셔츠 갈아입기 게임

투빅스에서 전 기수 엠티를 가서 대강당에서 게임을 한다. 총  $M \times N$ 명의 투빅이들은 빨간 티셔츠 아니면 파란 티셔츠를 입고 있다.  $M \times N$  행렬의 형태로  $M \times N$ 개의 의자가 놓여져 있는 강당에서 투빅이들은 무작위로 앉아 있다.

### [게임 룰]

1. 64명의 투빅이들은  $8 \times 8$  정사각형 형태로, 서로 다른 색깔의 티셔츠를 입고 번갈아 앉아있어야 한다. (이때,  $8 \times 8$  정사각형의 위치에는 제약이 없다.  $M \times N$  행렬 내에만 들어가있으면 된다.)
2.  $8 \times 8$  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  $8 \times 8$  정사각형 형태로 앉아 있기 위해 티셔츠를 갈아입어야 하는 최소 횟수(명수)를 출력하시오.

# Unit 04 | 3주차 과제 소개



17기 정규세션  
TOBIG'S 16기 이예림

## [입력 조건]

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

## [출력 조건]

- 첫째 줄에, 64명의 투빅이들이 알맞게 티셔츠를 입고  $8 \times 8$  정사각형 형태로 앉아 있기 위해 티셔츠를 갈아입어야 하는 최소 횟수(명수)를 출력하시오.

[입력 예시]

10 13

RRRRRRRRRBRBRB  
RRRRRRRRRBRBR  
RRRRRRRRRBRBRB  
RRRRRRRRRBRBR  
RRRRRRRRRBRBRB  
RRRRRRRRRBRBR  
RRRRRRRRRBRBR  
RRRRRRRRRBRBRB  
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 |



## 문제 4. 일곱 투빅이들

투빅스의 면접을 뚫고 총 7명의 투빅이들이 뽑혔다.  
그런데 정모를 했을 때 9명의 투빅이들이 모였다.

9명의 투빅이들은 모두 자신이 투빅스의 일원이라고 주장했다.  
다행스럽게도 일곱 투빅이들의 키의 합이 1000이 됨을 기억해 냈다.

아홉 투빅이들의 키가 주어졌을 때,  
진짜 일곱 투빅이들을 찾는 프로그램을 작성하시오.

### [입력 조건]

아홉 개의 줄에 걸쳐 투빅이들의 키가 주어진다.  
주어지는 키는 1000을 넘지 않는 자연수이며,  
아홉 투빅이의 키는 모두 다르며,  
가능한 정답이 여러 가지인 경우는 없다.

### [출력 조건]

일곱 투빅이의 키를 오름차순으로 출력한다.  
일곱 투빅이를 찾을 수 없는 경우는 없다.

[입력 예시]

200

70

230

190

100

150

250

80

130

[출력 예시]

70

80

100

130

190

200

230





## 문제 5. 단어 분할

알파벳 소문자로 이루어진 단어를 가지고 아래와 같은 과정을 해 보려고 한다.

먼저 단어에서 임의의 두 부분을 골라서 단어를 쪼갬다. 즉, 주어진 단어를 세 개의 더 작은 단어로 나누는 것이다. 각각은 적어도 길이가 1 이상인 단어여야 한다. 이제 이렇게 분할한 세 개의 작은 단어들을 앞뒤를 뒤집고, 이를 다시 원래의 순서대로 합친다.

예를 들어,

- 단어: tobigs
- 세 단어로 분할: to/bi/gs
- 각각 뒤집기: ot/ib/sg
- 합치기: otibsg

단어가 주어지면, 이렇게 만들 수 있는 단어 중에서 사전순으로 가장 앞서는 단어를 출력하는 프로그램을 작성하시오.



## 문제 5. 단어 분할

### [입력 조건]

첫째 줄에 영어 소문자로 된 단어가 주어진다.  
길이는 3이상 50이하이다.

### [입력 예시]

tobigs -> tob/ig/s

### [출력 조건]

첫째 줄에 구하고자 하는 단어를 출력하면 된다.

### [출력 예시]

botgis



## 3주차 알고리즘 과제

### 3주차 과제 Mimir

Course Code : f5575476a8 또는

Course Link : <https://class.mimir.io/courses/f5575476a8/registrations/new>

1주차 알고리즘 과제 기한 : **2월 14일 23:59 (2주)**

## 잊지 마세요!

투빅스 채점 사이트는 꼭! 가입 해주세요.

: 가입 방법 및 사용 가이드 <https://tobigs.gitbook.io/algorithm/>

코드와 링크는 매 과제마다 달라집니다.

질문, 힌트, 오류 문의는 지윤혁, 이기정, 이정우, 권소영에게 부탁드립니다!  
언제나 환영입니다 ^^\*



17기 정규세션  
TOBIG'S 16기 이예림

