



[A반] 3회차 과목평가 - 알고리즘

| Background

- ✓ 1차원 및 2차원 배열에 대해 이해하고 활용
- ✓ 기본적인 정렬 방법에 대해 이해하고 활용

| Goal

- ✓ 문제를 정확히 이해하고 알고리즘을 설계할 수 있다.
- ✓ 배열의 인덱스 연산을 통해 문제를 해결할 수 있다.

| 환경 설정

1) Pycharm(Python3.5이상)을 이용해서 코드를 작성하고 결과를 확인한다.

- 새로운 Pycharm 프로젝트를 생성 후 코드를 작성한다.

2) 파일 이름 및 제출 방법

- 1,2번 문제에 대한 소스 파일은 Algo문제번호_지역_반_이름.py로 만든다.
- 3번은 텍스트 파일로 작성한다.

Algo1_서울_1반_이싸피.py

Algo2_서울_1반_이싸피.py

Algo3_서울_1반_이싸피.txt

- 위 3개의 파일만 지역_반_이름.zip으로 압축하여 제출한다.

서울_1반_이싸피.zip

(탐색기에서 파일 선택 후 오른쪽 클릭 - 보내기 - 압축(zip)폴더 선택)
(edu.ssafy.com 사이트에 업로드)

3) 채점

- 주석이 없으면 해당 문제는 0점 처리될 수 있다.
- import를 사용한 경우 해당 문제는 0점 처리될 수 있다.
- 테스트 케이스별로 부분 채점 된다.

4) 테스트케이스는 부분적으로 제공되며, 전체가 공개되지는 않는다.

5) 각 문제의 배점이 다르므로 표기된 배점을 반드시 확인한다.

1번 문제(70점), 2번 문제(20점), 3번 문제(10점)

성실과 신뢰로 테스트에 볼 것 (부정 행위시 강력 조치 및 근거가 남음)

※ 소스코드 유사도 판단 프로그램 기준 부정 행위로 판단될 시, 0점 처리 및 학사 기준에



[A반] 3회차 과목평가 - 알고리즘

| 문제1 : 정원에 나무심기 (배점 70점)

김싸피는 집의 정원에 나무를 심으려고 한다. 정원은 사각형 모양이며 나무는 세로 줄로 심을 예정이다. 정원의 가장 왼쪽 세로 줄부터 나무를 심어야 하며 앞으로 나무가 커질 것을 고려하여 한 줄씩 띄어 심으려고 한다. 각 위치에 심을 나무의 가격은 알고 있다고 가정할 때 정원에 나무를 심기 위한 총 비용과 심은 나무의 수를 구해보자. 또한 심은 나무 중 가장 비싼 나무의 가격과 해당 나무의 열 번호를 계산해보자. 만약 가장 비싼 나무가 여러 개 심어져 있는 경우 가장 큰 열의 번호를 계산한다.

예를 들어 정원의 크기가 3 X 3이고 각 위치에 심을 나무의 가격이 다음 그림과 같다면 나무를 심는 총 비용은 $5+7+4+3+8+5 = 32$ 가 되고 심은 나무의 수는 6이 된다. 또한 가장 비싼 나무의 가격은 8이며 해당 나무가 심어진 열은 3이 된다.

5	2	3
7	1	8
4	6	5

또 다른 예를 들어 보면, 다음 그림과 같이 정원 크기가 4 X 5 인 경우 나무를 심는 총 비용은 $3+9+7+5+4+7+2+8+8+5+9+6 = 73$ 이 되고 심은 나무의 수는 12가 된다. 또한 가장 비싼 나무의 가격은 9이며 해당 나무는 1열과 5열에 심어져 있지만 열 번호가 더 큰 5가 계산된다.

3	2	4	1	8
9	5	7	6	5
7	3	2	5	9
5	1	8	4	6

정원 크기인 N X M 리스트가 주어질 때 나무를 심는 총 비용과 심은 나무의 수, 심은 나무 중 가장 비싼 나무의 가격, 가장 비싼 나무가 심어진 열을 출력하는 프로그램을 만드시오.



[A반] 3회차 과목평가 - 알고리즘

[입력]

첫 줄에는 테스트케이스 개수 T 가 주어진다. ($1 \leq T \leq 10$)

다음 줄부터 테스트 케이스의 첫 줄엔 정원의 크기인 행의 개수 N 과 열의 개수 M 이 주어진다. ($3 \leq N \leq 20, 3 \leq M \leq 20$)

그 다음 줄부터는 정원 영역이 N 줄에 걸쳐 각 행 별로 M 개의 자연수가 공백으로 구분되어 주어진다. 주어지는 자연수는 100이하이다.

[출력]

각 줄에 #과 1부터인 테스트케이스번호를 출력하고 나무를 심는 총 비용, 심은 나무의 수, 가장 비싼 나무의 가격, 가장 비싼 나무가 심어진 열을 빈칸으로 구분하여 출력하시오.

[입력 예시]

```
3
3 3
5 2 3
7 1 8
4 6 5
4 5
3 2 4 1 8
9 5 7 6 5
7 3 2 5 9
5 1 8 4 6
5 6
40 15 33 52 12 37
70 23 73 12 34 54
23 10 37 15 45 80
50 60 48 24 19 20
55 28 32 66 80 42
```

[출력 예시]

```
#1 32 6 8 3
#2 73 12 9 5
#3 651 15 80 5
```



[A반] 3회차 과목평가 - 알고리즘

| 문제 2 : 보드게임 (배점 20점)

A와 B가 보드게임을 하려고 한다. 게임의 룰은 아래와 같다.

- 보드의 총 길이는 20칸이다.
- 게임을 시작하면 A,B의 말은 모두 첫번째 칸에 위치시킨다.
- 주사위를 굴러 나온 숫자만큼 이동을 한다.
- 이동 후 상대방의 말이 그 위치에 있으면 상대방의 말을 잡는다. (상대방의 말을 처음위치로 이동시킨다. 말을 잡더라도 한번 더 주사위를 던지지 않는다.)
- 20번째 칸에 먼저 도착하는 사람이 승리한다. (넘어가도 도착한 것으로 본다.)
- 각각 10번의 주사위를 굴러 아무도 도착하지 못했다면 무승부로 게임이 종료된다.

게임을 진행 후 승자가 누구인지 출력하시오.

[입력]

가장 첫 줄에는 테스트 케이스의 개수 T가 주어지고, 그 아래로 각 테스트 케이스가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 첫번째로 주사위를 굴리는 사람이 누구인지 주어진다.

두번째 줄에는 A의 주사위 숫자가 순서대로 주어진다.

세번째 줄에는 B의 주사위 숫자가 순서대로 주어진다.

[출력]

출력의 각 줄은 '#t'로 시작하고, 공백을 한 칸 둔 다음 승자를 출력한다.

단, 무승부인 경우 AB를 출력한다.

(t는 테스트 케이스의 번호를 의미하며 1부터 시작한다.)

[A반] 3회차 과목평가 - 알고리즘



[입력예시]

```
3
A
6 6 6 6 6 6 6 6 6 6
1 1 1 1 1 1 1 1 1 1
B
2 3 2 3 2 3 2 3 2 3
3 2 3 2 3 2 3 2 3 2
A
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

[출력예시]

```
#1 A
#2 A
#3 AB
```



[A반] 3회차 과목평가 - 알고리즘

| 문제 3 : 선택 정렬 (배점 10점)

다음과 같은 배열 arr이 있다.

arr	5	2	6	1	9	3	7	8	4
-----	---	---	---	---	---	---	---	---	---

arr을 오름차순으로 선택 정렬하려 한다.

1. 선택 정렬을 위한 슈도코드(pseudo code) 를 완성하시오.
단, min(), max(), sort() 등의 내장함수를 사용해 슈도코드를 표현해서는 안된다.

N : 정렬하려는 배열의 길이, arr : 정렬하려는 대상 배열

selection_sort(N, arr):

2. 선택 정렬은 배열의 요소 간 교환이 일어나며 배열을 정렬하는 알고리즘이다. 교환이 일어났을 때 배열의 모양을 모두 나열하고, 교환이 일어나는 횟수를 적으시오. (단, 교환이란 서로 다른 두 요소 간의 위치 변경을 의미한다.)
예시)

- 1 2 6 5 9 3 7 8 4 1,5 교환