

# [A반] 3회차 과목평가 – 알고리즘 기본



## | Background

- ✓ 배열에 대해 이해하고 활용
- ✓ 고지식한 알고리즘을 이해하고 활용
- ✓ 정렬 알고리즘을 이해하고 활용

## | Goal

- ✓ 문제를 정확히 이해하고 알고리즘을 설계할 수 있다.
- ✓ 고지식한 접근방법을 통해 문제를 해결할 수 있다.
- ✓ 정렬 알고리즘을 설명할 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 없는 경우, 주석이 코드 내용과 맞지 않는 경우, 지정된 출력 형식을 만족하지 않는 경우 해당 문제는 0점 처리될 수 있다.

- import를 사용한 경우 해당 문제는 0점 처리될 수 있다. (import sys도 예외 없음)

4) 테스트케이스는 부분적으로 제공되며, 전체가 공개되지는 않는다.

5) 각 문제의 배점이 다르므로 표기된 배점을 반드시 확인한다.

- 1번 50점, 2번 30점, 3번 20점

- ## 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

의거 조치 실시 예정

## [A반] 3회차 과목평가 – 알고리즘 기본



## | 문제 1 : 구간별 최대 최소 구하기 (배점 50점)

N개의 정수가 들어 있는 배열에서 이웃한 M개의 숫자 중 가장 큰 값과 작은 값을 찾아 차이를 출력하시오.

다음은 N=5, M=3이고 5개의 숫자 1 2 3 4 5가 배열 v에 들어있는 경우이다.

예를 들어,

1	2	3	4	6
---	---	---	---	---

이웃한 수 [1,2,3]의 최대값은 3, 최소값은 1이므로 차이 2를, [2,3,4]의 최대값은 4, 최소값은 2이므로 2를, [3, 4, 6]은 최대값 6, 최소값 3으로 3을 출력한다.

[제약사항]

파이썬에서 제공하는 내장함수 max, min은 사용하지 않는다.

[입력]

첫 줄에 테스트케이스 수가 주어진다.

다음 줄부터 테스트케이스의 첫 줄에 정수의 개수 N과 구간의 개수 M이 주어진다. ( $5 \leq N \leq 20$ ,  $2 \leq M < N$ )

다음 줄에 N개의 정수  $a_i$ 가 주어진다. ( $1 \leq a \leq 10000$ )

[출력]

각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 구간 별 최대값과 최소값의 차이를 출력한다.

# [A반] 3회차 과목평가 – 알고리즘 기본



[입력 예시]

```
3
5 3
1 2 3 4 6
5 3
8145 5300 1370 9098 7331
6 3
603 5178 8071 7796 5917 1282
```

[출력 예시]

```
#1 2 2 3
#2 6775 7728 7728
#3 7468 2893 2154 6514
```

## [A반] 3회차 과목평가 – 알고리즘 기본



## | 문제2 : 토지 개발 (배점 30점)

싸피 나라는 신도시를 건설하기 위한 계획을 추진 중이다. 신도시를 건설하기 위해서는 사각형으로 된 평평한 토지가 필요하다.

따라서, 신도시 예정지가 결정되면 토지를 평탄화하는 작업이 필요하다.

그림1은 3 x 3 크기의 토지 정보이다. 1 x 1 영역의 정수 값은 토지의 높이에 해당하는 값이다.

&lt;그림1&gt;

2	1	1
3	5	1
3	1	2

평탄화는 모든 곳의 높이를 동일하게 하는 작업이다. 평탄화 높이는 모든 높이 값들의 평균으로 결정된다. 그림1의 평탄화 높이는 총합(=19)을 개수(9)로 나눈 2가 된다. 나누기의 결과는 소수점 이하는 무시한 정수가 된다.

평탄화 높이가 결정되면 평탄화 작업에 필요한 비용을 계산할 수 있다. 1 x 1 영역을 평탄화 하기 위해서는 평균 높이와 해당 영역 높이와의 차이 값 만큼 비용이 필요하다.

아래 그림 2는 평탄화 높이가 2일 때 원래 높이와의 차이를 보여준다. 평탄화 비용은 아래 오른쪽 그림의 차이 값들의 총합인 9가 된다.

&lt;그림2&gt;

2	1	1
3	5	1
3	1	2

0	1	1
1	3	1
1	1	0

원래 높이와 평균과의 차이

N x N 형태의 큰 지도 정보와 신도시 개발지인 사각 영역을 의미하는 두개의 좌표 값이 주어질 때 평탄화 높이와 비용을 계산하는 프로그램을 작성해보자. 사각 영역에 해당하는 두 좌표는 좌상단과 우하단의 좌표이다.

## [A반] 3회차 과목평가 – 알고리즘 기본



다음은 6 x 6 영역의 지도 정보이고, 신도시 개발지는 색칠된 영역이다.  
개발지의 좌상단, 우하단 좌표는 (1, 2) 와 (3, 4)이다. 개발지의 평탄화 높이는 2 이고 비용은 9 이다.

	0	1	2	3	4	5
0	1	2	3	1	1	1
1	1	2	2	1	1	2
2	2	2	3	5	1	3
3	3	3	3	1	2	2
4	1	2	1	2	2	1
5	2	3	1	2	3	1

## [입력]

- 첫 줄에 테스트케이스 수가 주어진다.
- 다음줄에 지도의 크기  $N(6 \leq N \leq 20)$ 이 주어진다.
- 다음 줄에 사각 영역의 좌상단과 우하단 좌표에 해당하는 네 개의 값이 주어진다. 먼저 나오는 두 정수가 좌상단 좌표의 행과 열의 위치에 해당한다. 좌표 값은 0이상  $N-1$  이하의 값이다.
- 그 다음 줄부터 지도 정보가  $N$ 개의 줄에 공백으로 구분된  $N$ 개의 값들로 주어진다. 높이 값들은 1 이상 9이하의 값들이다.

## [출력]

- '#' 과 테스트 케이스 번호를 출력하고 평탄화 높이와 평탄화 비용을 공백으로 구분해서 출력한다.

입출력 예시는 다음 페이지에 있습니다.

# [A반] 3회차 과목평가 – 알고리즘 기본



[입력 예시]

```
3
6
1 2 3 4
1 2 3 1 1 1
1 2 2 1 1 2
2 2 3 5 1 3
3 3 3 1 2 2
1 2 1 2 2 1
2 3 1 2 3 1
6
0 2 4 3
1 1 5 1 5 5
1 4 2 2 1 3
3 1 1 4 5 5
2 4 3 1 2 2
2 3 1 4 5 4
4 3 5 3 3 5
7
1 0 3 6
1 5 2 2 2 3 5
1 6 6 4 6 5 5
6 4 2 2 4 1 5
3 5 2 6 5 3 2
6 4 5 6 3 3 3
3 5 2 3 2 5 1
1 5 1 1 3 4 4
```

[출력 예시]

```
#1 2 9
#2 2 12
#3 3 36
```

## [A반] 3회차 과목평가 – 알고리즘 기본



### | 문제3 : 정렬 (서술형, 배점 20점)

※ 다음과 같은 배열 a가 있다. 다음 물음에 답하시오.

55	07	78	12	42
----	----	----	----	----

1. 버블정렬을 이용하여 내림차순으로 정렬하는 함수를 완성하시오.

# a : 정렬할 배열, n : 배열의 크기  
def bubble\_sort(a, n):

2. 위의 배열 a를 버블정렬을 이용하여 내림차순으로 정렬하려고 한다. 교환이 일어났을 때 배열의 모양을 모두 나열하고 교환이 일어나는 횟수를 쓰시오. (단, 교환이란 서로 다른 두 요소 간의 위치변경을 의미한다)

예시)

- 55 78 7 12 42      78, 7 교환