

[Python트랙] 월말평가2 – 알고리즘 기본



| Background

- ✓ 배열에 대한 이해와 사용
- ✓ 반복문에 대한 이해와 사용
- ✓ 스택에 대한 이해와 사용

| Goal

- ✓ 인덱스 연산을 통해 배열의 요소에 접근할 수 있다.
- ✓ 스택을 이해하고 활용할 수 있다.

| 환경 설정

1) Pycharm(Python3.7이상)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.
- pypy의 경우 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출 시 아래와 같이 변경한다.
- 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)

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 40점, 2번 35점, 3번 25점

성실과 신뢰로 테스트에 임할 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

[Python트랙] 월말평가2 – 알고리즘 기본



| 문제 1 : 평지 만들기 (배점 40점)

김싸피는 전체 크기가 $N \times N$ 인 마당 중 일부를 평지로 만들려고 한다. 이 때, 평지 만들기를 완료하기 위해서 몇 번의 작업을 해야 하는지 구하는 프로그램을 작성하라.

[제약사항]

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	3	2	4
2	0	3	2	6	1
3	0	4	1	2	5
4	0	0	0	0	0

평지로 만들려고 하는 평탄화 영역은 직사각형 형태로 왼쪽 위, 오른쪽 아래 칸의 좌표가 주어진다. (위 예시의 녹색 영역은 1 1 3 4로 주어진다.)

평지를 만들기 위해서 김싸피는 해당 영역 높이의 **평균값**으로 평탄화 작업을 하려고 한다.

***평균값은 (평탄화 영역의 높이 값의 합) // (영역의 칸 수) 이다.** (소수점 아래는 버린다.)

한 번의 작업으로 한 칸의 높이를 1 높이거나 1 낮게 할 수 있다. 예를 들어 기존 높이가 4인 칸은 높이 2로 만드는데 2회의 작업이 필요하다.

위 예시에서 녹색인 평탄화 영역의 총합은 34, 영역의 넓이는 12 이므로 영역의 평탄화 높이는 $34 // 12$ 인 2이다. 따라서 해당 영역을 모두 높이 2로 평지를 만들기 위해서 **총 16번**의 작업이 필요하다.

[Python트랙] 월말평가2 – 알고리즘 기본



[입력]

첫 줄에 테스트케이스 수가 주어진다.

각 테스트 케이스의 첫 줄에 N 주어진다.

이후 평지로 만들려는 영역의 좌측상단 행과 열 번호와 우측하단 행과 열 번호가 순서대로 띄어쓰기로 구분되어 주어진다. (좌표는 0부터 시작한다.) 앞의 예시의 경우 '1 1 3 4' 가 좌표로 주어진다.

이후 N 줄에 걸쳐 N 개의 정수 A_i 가 주어진다.

($3 \leq N \leq 10$, $0 \leq A_i \leq 100$)

[출력]

각 줄마다 " #T " (T는 테스트 케이스 번호)를 출력한 뒤, 빈칸에 이어 작업의 횟수를 출력한다.

[입력 예시]

```
3
5
1 1 3 4
0 0 0 0
0 1 3 2 4
0 3 2 6 1
0 4 1 2 5
0 0 0 0 0
5
0 0 4 4
4 3 3 4 3
3 1 3 2 4
5 3 2 6 1
3 4 1 2 5
3 3 3 3 3
3
0 0 0 2
1 1 1
1 1 1
1 1 1
(algo1_sample_in.txt 참고)
```

[출력 예시]

```
#1 16
#2 20
#3 0

(algo1_sample_out.txt 참고)
```

[Python트랙] 월말평가2 – 알고리즘 기본



| 문제 2 : 같은 모양 찾기 (배점 35점)

1과 0으로 표시된 NxN 크기의 2차원 배열이 있다.

3x3 크기의 입력 받은 패턴의 개수를 구하는 프로그램을 작성하시오.

단, 다음 조건에 따라 개수를 구해야 한다.

- (1) 3x3 패턴과 9개 값이 모두 일치해야 해당 패턴을 찾은 것으로 계산한다.
- (2) 이전에 찾은 모양과 일부 영역이 겹치는 경우도 카운트한다.

아래 6x6 배열에서 우측의 3x3 패턴은 총 3개가 존재한다.

1	1	1	1	1	1
0	1	0	1	0	1
1	1	1	1	1	1
0	0	1	0	1	0
1	1	0	1	1	1
1	0	1	1	1	1

1	1	1
0	1	0
1	1	1

[입력]

첫 줄에 테스트케이스의 개수 T가 주어진다. 다음 줄부터 케이스 별로, 첫 줄에 N (N은 10이하의 자연수), 그 다음 줄에 N개씩 N개의 줄에 걸쳐 0 또는 1값이 공백으로 구분되어 주어진다.

이후 0또는 1의 3개의 숫자가 3줄에 걸쳐 공백으로 구분되어 주어진다.

[출력]

#과 1번부터인 테스트케이스 번호와 답을 공백으로 구분하여 출력한다.

[Python트랙] 월말평가2 – 알고리즘 기본



[입력 예시]

```
2
4
1 1 1 1
0 1 0 0
0 1 0 1
1 0 1 0
1 1 1
0 1 0
0 1 0
6
1 1 1 1 1 1
0 1 0 1 0 1
1 1 1 1 1 1
0 0 1 0 1 0
1 1 0 1 1 1
1 0 1 1 1 1
1 1 1
0 1 0
1 1 1
```

(algo2_sample_in.txt 참고)

[출력 예시]

```
#1 1
#2 3
```

(algo2_sample_out.txt 참고)

[Python트랙] 월말평가2 – 알고리즘 기본



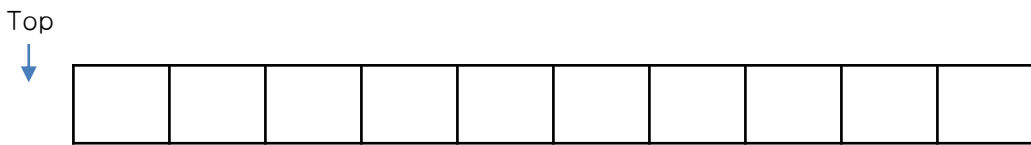
| 문제 3 : 스택 (배점 25점)

스택은 물건을 쌓아 올리듯 자료를 쌓아 올린 형태의 자료구조이다. Top은 스택의 맨 위 원소의 위치를 나타낸다.

3-1. 스택의 특성을 간단히 설명 하시오.

3-2. 다음과 같이 빈 스택이 있다. 여기에 주어진 연산이 이뤄 지고 난 후, 가장 먼저 저장된 값부터 Top 이 가리키는 원소까지 순서대로 적으시오.

(두 번째 push까지의 예) AB



push(A)

push(B)

pop()

push(D)

push(C)

push(F)

push(E)

pop()

push(A)