

[Python트랙] 과목평가3 - 알고리즘 기본



| Background

- ✓ 배열에 대한 이해와 활용
- ✓ 스택에 대한 이해와 활용

| Goal

- ✓ 반복문과 조건문을 이용하여 배열의 요소에 접근할 수 있다.
- ✓ 문제의 조건을 정확히 이해하고 해결할 수 있다.
- ✓ 스택의 동작을 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 폴더, 프로젝트, 파일이름에 한글을 사용할 수 없으므로 algo1.py, algo2.py 로 만들고 제출시 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 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

[Python반] 과목평가3 – 알고리즘 기본



| 문제 1 : 봉우리(배점 40점)

[제약사항] 파이썬 내장함수 `max`, `min`, `sum`은 사용할 수 없다.

싸피 등산회를 위해 $N \times N$ 인 2차원 배열 형태의 지도 정보가 주어진다. 각 셀의 값은 해당 지역의 높이 값이다. 어떤 지역의 높이가 인접지역의 높이보다 크면 그 지역을 **봉우리**라 한다.

인접지역은 상하좌우 4곳이며, 지도의 가장자리는 인접한 지역이 3곳이나 2곳 일 수 있다.

그림 1은 4×4 지도이고, 봉우리는 주황색으로 칠해진 높이 3인 지역이다. 인접한 상/하는 높이가 2이고 좌/우는 1 이다.

1	1	2	1
2	1	3	1
2	1	2	1
3	4	4	4

<그림 1>

2	3	2	1	3
2	2	1	3	2
2	2	2	3	2
1	3	3	3	1
1	4	2	2	2

<그림 2>

그림 2의 (0행, 1열)의 높이는 3이고, 인접 지역인 왼쪽, 오른쪽, 아래의 높이가 모두 2라서 산봉우리가 된다. (0행, 4열)과 (4행, 1열)도 인접 의 높이보다 높으므로 봉우리가 된다.

그림 2의 지도에서 봉우리는 주황색으로 칠해진 3곳 이다.

지도가 주어지면 봉우리에 해당하는 모든 지역을 찾아보자.

[Python반] 과목평가3 – 알고리즘 기본



[입력]

- 첫 줄에 지도의 개수가 입력된다.
- 테스트 케이스의 첫 줄에는 지도의 가로 세로 크기 N이 주어진다. ($4 \leq N \leq 30$)
- 다음 N 개의 줄에 걸쳐 공백으로 구분된 N 개의 높이 h가 주어진다. ($0 \leq h \leq 20$)

[출력]

- '#테스트케이스 번호' 를 출력하고, 봉우리에 해당하는 지역의 수를 출력한다. **봉우리가 없는 경우는 0을 출력한다.**

[입력 예시]

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

[출력 예시]

```
#1 1
#2 0
#3 3
(algo1_sample_out.txt 참고)
```

[Python반] 과목평가3 – 알고리즘 기본



| 문제 2 : 괄호 연산 (배점 35점)

[제약 사항] 내장함수 `sum()`, 인덱스 슬라이싱은 사용할 수 없다.

다음과 같이 배열에 0이상인 정수 A_i , 소괄호 `()`, 중괄호 `{}`가 들어있다.

(2	{	(5	8)	7	}	4	3)
---	---	---	---	---	---	---	---	---	---	---	---

괄호안의 숫자는 다음 규칙에 따라 연산을 한다.

- 가장 안쪽의 괄호를 먼저 연산한다.
- 소괄호 `()` 안의 숫자는 모두 더한다.
- 중괄호 `{}` 안의 숫자는 모두 곱한다.
- 괄호의 짝이 맞지 않는 경우 -1을 출력한다.

위의 배열에서 연산은 다음과 같이 이루어진다.

$(5 + 8)$ 계산 후 `(2 { 13 7 } 4 3)`

$\{13 * 7\}$ 계산 후 `(2 91 4 3)`

$(2 + 91 + 4 + 3)$ 계산 후 100을 출력한다.

다음과 같이 괄호의 짝이 맞지 않는 경우 -1을 출력한다.

`(2 { (5 8 } 7 } 4 3)`

`(5 8`

`5 8)`

`) 5 8 (`

괄호에 둘러 쌓이지 않는 숫자가 있는 경우에도 -1을 출력한다.

`1 (2 3) 4`

[Python반] 과목평가3 – 알고리즘 기본



괄호안에 들어갈 수식이 문자열로 주어지면, -1 또는 정수를 출력하는 프로그램을 만드시오.

괄호안에 숫자가 하나만 있는 경우의 연산은, 값은 변하지 않고 괄호만 없어진다.

(2) == 2, {3} == 3 이 된다.

괄호검사 결과만 정확하게 출력해도 부분점수를 받을 수 있으므로 최선을 다해야한다.

[입력]

첫 줄에 테스트케이스 수 T가 주어진다.

다음 줄부터 테스트케이스 별로 한 줄에 배열에 들어갈 수식이 공백없이 주어진다. 처음 주어지는 숫자 A_i 는 모두 0이상 9이하인 한자리 정수이다. ($0 \leq A_i \leq 9$), 수식의 길이는 1이상 100이하이다.

[출력]

#과 테스트케이스 번호, 빈칸에 이어 답을 출력한다.

[입력예시]

```
3
(2{(58)7}43)
(2{(58}7)43)
2{(58)7}43
(algo2_sample_in.txt 참고)
```

[출력예시]

```
#1 100
#2 -1
#3 -1
(algo2_sample_out.txt 참고)
```



| 문제 3 : (배점 25점)

(1) 스택(stack)에 대해 간단히 설명하시오.

(2) Depth First Search(DFS, 깊이우선탐색)에 대해 간단히 설명하시오.

(3) 다음과 같은 그래프에서 1번 정점부터 DFS를 시작하는 경우, 정점의 방문순서는 1 2 4 6 5 7 3 이다. 같은 규칙으로 4번 정점부터 DFS를 시작하는 경우의 방문순서를 적고, 갈림길에서 어떤 선택을 했는지 간단히 설명하시오.

