



## 40. 정수 삼각형

### 문제 설명

```

    7
   3 8
  8 1 0
 2 7 4 4
4 5 2 6 5
```

- 위와 같은 삼각형의 꼭대기에서 바닥까지 이어지는 경로 중, 거쳐간 숫자의 합이 가장 큰 경우 찾기
- 아래 칸으로 이동할 때는 대각선 방향으로 한 칸 오른쪽 또는 왼쪽으로만 이동 가능
- 예를 들어 3에서는 그 아래칸의 8또는 1로만 이동이 가능
- 삼각형 정보가 담긴 배열 triangle이 매개변수로 주어짐
- 거쳐간 숫자의 최대 값을 return함

### 제한 사항

- 삼각형의 높이는 1 이상 500 이하
- 삼각형을 이루고 있는 숫자는 0 이상 9,999 이하의 정수

### 입출력 예

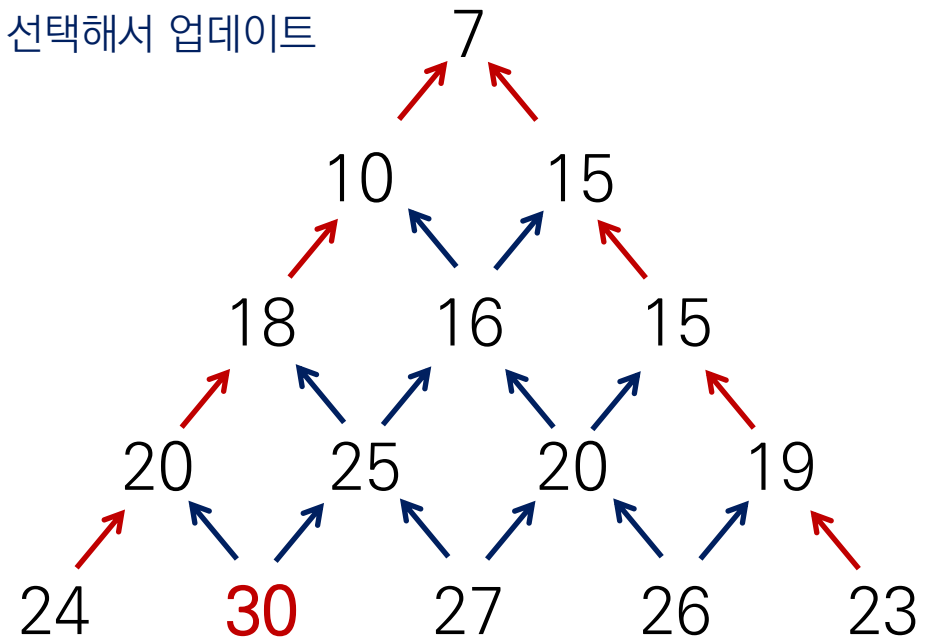
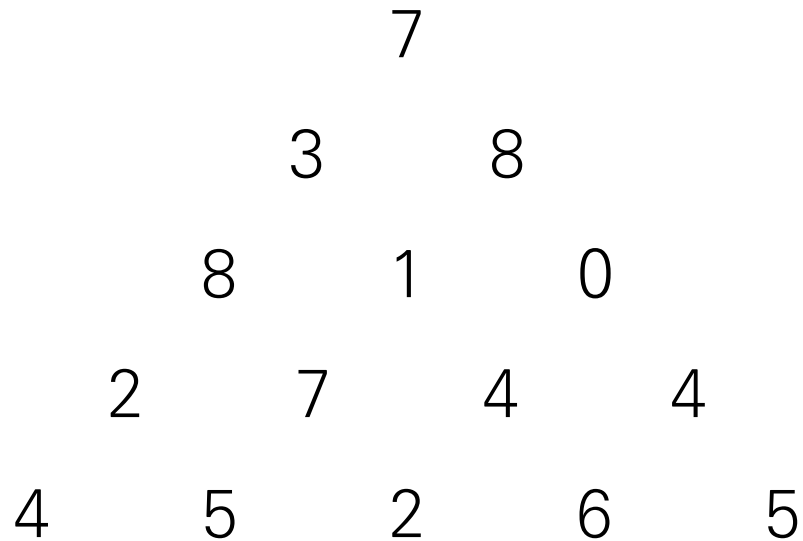
triangle	result
[[7], [3, 8], [8, 1, 0], [2, 7, 4, 4], [4, 5, 2, 6, 5]]	30



## 40. 정수 삼각형

### 문제 풀이

- 꼭대기부터 업데이트 해가면서 풀어야 하는 문제
- 한 칸을 업데이트를 할 경우에는 두 가지 경우가 있음
  1. 양쪽 끝일 때 : 양쪽 끝의 숫자들은 그 전 줄의 처음 혹은 끝의 영향만 받아 업데이트
  2. 그 외의 경우 : 영향을 주는 2개의 숫자를 비교해 둘 중 큰 경우 선택해서 업데이트





## 40. 정수 삼각형

### 코드 작성

```
def solution(triangle):
    answer = 0
    for i in range(1, len(triangle)): # 꼭대기는 업데이트가 필요 없기 때문에 2번 줄 부터 마지막 줄까지 반복
        for j in range(i+1): # 현재 줄의 각 요소에 대해 반복(즉, 요소 = i+1)
            if j==0: # 각 줄의 첫번째 요소는 1번 방식으로 전 줄의 첫번째 요소를 더해 업데이트
                triangle[i][j] += triangle[i-1][j]
            elif j==i: # 각 줄의 마지막 요소는 1번 방식으로 전 줄의 마지막 요소를 더해 업데이트
                triangle[i][j] += triangle[i-1][j-1]
            else: # 각 줄의 중간 요소는 2번 방식으로 자신에게 영향을 주는 두 수의 크기를 비교해 큰 값을 더해 업데이트
                triangle[i][j] += max(triangle[i-1][j-1], triangle[i-1][j]) # max 함수 : 주어진 값들 중에서 가장 큰 값을 찾는 함수
    answer = max(triangle[-1]) # 마지막 줄에서 가장 큰 수를 리턴 값에 저장
    return answer # 파이썬에서는 음수 인덱스를 사용하여 리스트의 끝에서부터 요소를 참조할 수 있음
```