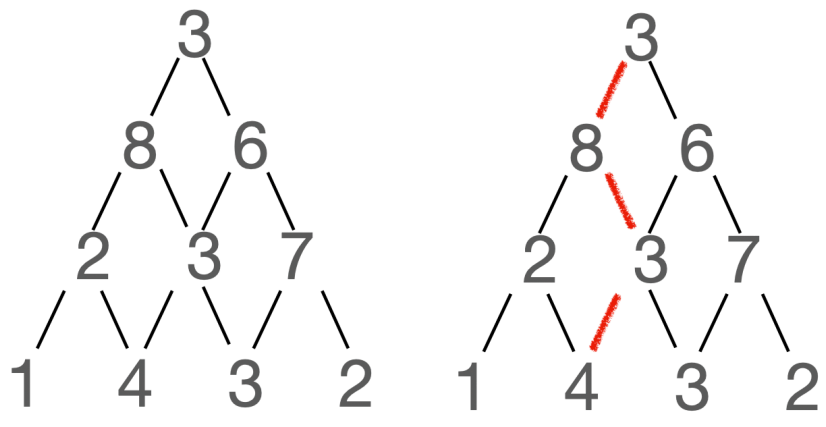


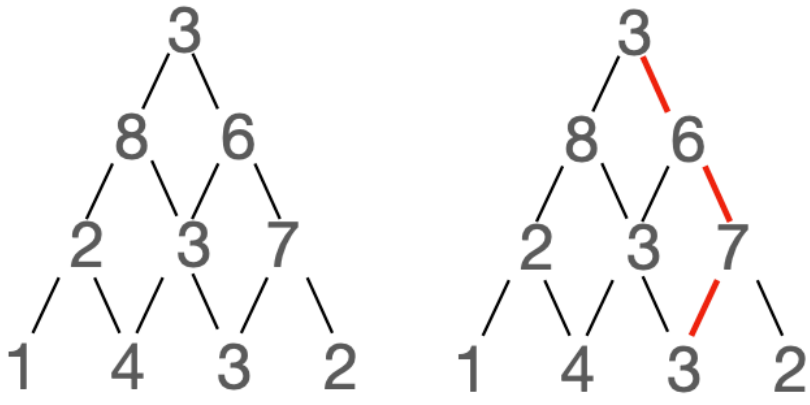
1. 함수 simple_max_path() 구현하기

주어진 숫자들의 꼭대기부터 시작해 아래방향으로 왼쪽, 오른쪽 중 하나를 선택해가며 맨 아래까지 내려가고자 한다. 길을 선택할 때, 둘 중 더 큰 수를 가진 쪽을 선택하고 그 지나온 숫자들을 모두 더하는 함수 simple_max_path()를 구현해보자.

함수명	simple_max_path(List[int])
입력	각 수가 담겨있는 2차원 리스트
출력	선택할 수 있는 수 중 더 큰 수를 선택하여 지나온 숫자들의 합
조건	<ul style="list-style-type: none"> - 현재 자리에서 다음 수를 고를 때, 더 큰 수를 greedy하게 선택 - 비교되는 수가 같을 경우, 오른쪽 수를 선택함 - 마지막 숫자까지 다다랐을 때 반복문 종료 - '예시 결과'의 코드와 똑같이 입력 시 같은 결과 나와야 함.
참고	
예시 결과	<pre>M = [[1,2,8,3],[0,4,3,6],[0,0,3,7],[0,0,0,2]] print('최댓값 : ',simple_max_path(M)) >> 최댓값 : 18</pre>

2. 함수 max_path() 구현하기

주어진 숫자들의 꼭대기부터 시작해 아래방향으로 왼쪽, 오른쪽 중 하나를 선택해가며 맨 아래까지 내려간다고 할때, 지나온 숫자들을 모두 더한 최종 값이 최대가 되도록 길을 선택한다. 이때, 지나온 수들의 최대합을 구하는 max_path() 함수를 구현해보자.

함수명	max_path(List[int])
입력	각 수가 담겨있는 2차원 리스트
출력	최댓값
조건	<ul style="list-style-type: none"> - 이전에 계산했던 값을 활용하여 다음 문제를 푸는 방식의 동적 프로그래밍으로 작성 - 입력값으로 받은 변수와 같은 크기의 차원의 새로운 리스트를 만들어 구현 - 이미 계산한 값을 다음 값을 만들때 활용할 수 있도록 하는 순서로 for 문을 돌면서 값을 업데이트함 - 다음 숫자를 선택할 때 둘 중 큰 수를 선택하는 것이 아닌, 총 합이 최대가 되도록 하는 수를 선택해야 함 - 필요시 파이썬 내장함수 max() 사용가능 - '예시 결과'의 코드와 똑같이 입력 시 같은 결과 나와야 함.
참고	
예시 결과	<pre>M = [[1,2,8,3],[0,4,3,6],[0,0,3,7],[0,0,0,2]] print('최댓값 : ',max_path(M)) >> 최댓값 : 19</pre>