

2022.12.30 스터디 자료

❄️ 내가 푼 문제 1

- LeetCode. 015 [3Sum](#)

❄️ 방법

```
class Solution {
    public static List<List<Integer>> threeSum(int[] nums) {

        /* 반환값 선언 */
        List<List<Integer>> result = new ArrayList<>();

        /* 배열 정렬 */
        Arrays.sort(nums);

        /* 반복문 돌기 -> i, left, right 세 개 인덱스의 값 더하기 */
        for (int i = 0; i < nums.length - 2; i++) {

            /* 만약 i와 i-1의 값이 같다면, 세 개를 더했을 때 0이 아니므로 아래의 코드가 아닌
            다음 i의 값을 확인하도록 continue; */
            if (i > 0 && nums[i] == nums[i-1]) continue;

            /* 세 값의 합이기 때문에 두 값을 구할 인덱스 지정 */
            int left = i + 1;
            int right = nums.length - 1;

            /* 반복문을 하나 더 열어서 확인해주기 */
            while (left < right) {

                /* nums 배열의 세 인덱스의 값을 더함 */
                int sum = nums[left] + nums[right] + nums[i];

                /* 만약 더한 값이 0이면, 결과를 반환할 배열리스트에
                세 값을 요소로 가진 배열 리스트를 더해줌 */
                if (sum == 0) {
                    result.add(Arrays.asList(nums[i], nums[left], nums[right]));
                    left++;
                    right--;

                    /*
                    * 비교하는 left와 right의 값이 중복이면, 인덱스만 바꿔준다
                    -> 세 값을 요소로 가진 배열 리스트 중복 방지
                    */
                    while (nums[left] == nums[left - 1] && left < right) left++;
                    while (nums[right] == nums[right + 1] && left < right) right--;
                } else if (sum > 0) {
                    right--;
                } else {
                    left++;
                }
            }
        }

        return result;
    }
}
```

❄️ 내가 푼 문제 2

- 프로그래머스. 정수 삼각형

▲ 주어진 삼각형

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

▲ DP 삼각형

	0	1	2	3	4
0	7				
1	10	15			
2	18	16	15		
3	20	25	20	19	
4	24	30	27	26	24

❄️ 방법

```
class Solution {
    public static int solution(int[][] triangle) {
        int answer = 0;

        int[][] dp = new int[triangle.length][triangle.length];
        dp[0][0] = triangle[0][0];

        for (int i = 1; i < triangle.length; i++) {

            /* 좌측 */
            dp[i][0] = dp[i - 1][0] + triangle[i][0];

            /* 우측 */
            dp[i][i] = dp[i - 1][i - 1] + triangle[i][i];

            /* 중앙 */
            for (int j = 1; j < i; j++) {
                dp[i][j] = Math.max(dp[i - 1][j], dp[i - 1][j - 1]) + triangle[i][j];
            }
        }

        for (int i = 0; i < triangle.length; i++) {
            answer = Math.max(answer, dp[triangle.length - 1][i]);
        }

        return answer;
    }
}
```