

유형	Brute Force	소요 시간	50분										
문제	<div><div>[문제 설명]</div><div>아파트 단지에 배송구역을 나누려 합니다. 단지에는 배송이 가능한 구역과 놀이터와 같이 배송할 수 없는 구역이 있으며, 배송원의 근무 피로도를 고려하여 각 배송원이 맡은 구역은 인접해야 합니다.</div><div>아파트 단지의 크기 $h \times w$가 주어질 때, 배송구역을 나누는 방법의 수를 구하는 함수, solution을 완성해주세요.</div><div>[제한 사항]</div><div><div>- 모든 배송원은 세 구역을 맡아야 합니다.</div><div>- 배송구역이 겹치거나, 배송할 수 없는 구역을 맡을 수 없습니다.</div><div>- 배송원은 인접한 구역만 맡을 수 있으며, 인접함은 상하좌우로 붙어있는 구역을 말합니다.</div><div>- 배송이 가능한 구역은 30을 넘지 않습니다.</div></div><div>[입력 형식]</div><div><div>- 두 자연수 h와 w는 1 이상 50 이하의 자연수입니다.</div><div>- 공백이 없는 $h * w$개의 글자로 단지 모양이 주어집니다.</div><div>(글자는 .과 @로 구성되어 있으며, .은 배송이 가능한 지역, @는 배송이 불가능한 지역을 나타냅니다.)</div></div><div>[출력 형식]</div><div>주어진 아파트 단지의 배송구역을 나누는 방법의 수를 출력합니다.</div></div>												
매개변수 & 리턴타입	<table><tr><th colspan="2">매개변수</th><th>Return</th></tr><tr><td>number</td><td>h</td><td rowspan="3">string</td></tr><tr><td>number</td><td>w</td></tr><tr><td>string</td><td>areaText</td></tr></table>			매개변수		Return	number	h	string	number	w	string	areaText
매개변수		Return											
number	h	string											
number	w												
string	areaText												
초기 코드	<pre>/** * @param h {number} * @param w {number} * @param areaText {string} * @return {number} */ function solution (h, w, areaText) { var answer = 0; return answer; }</pre>												
모범 답안	<pre>var solution = function (h, w, areaText) { var area = makeArea(h, w, areaText); if (area === null) { return 0; } }</pre>												

```

    return countDeliveryCase(area);
};

var makeArea = function (h, w, areaText) {
    // area 배열 생성
    var area = [];

    var deliveryAvailableNum = 0;
    for (var i = 0; i < h; i++) {
        // area 이중 배열 생성
        area[i] = [];

        // area 값 설정
        for (var j = 0; j < w; j++) {
            var index = (w * i) + j;
            var token = areaText[index];
            var value = token === '.' ? 0 : 1;
            area[i].push(value);

            if (token === '.') {
                deliveryAvailableNum++;
            }
        }
    }

    var isResolvable = deliveryAvailableNum % 3 === 0;
    return (isResolvable) ? area : null;
}

var countDeliveryCase = function (area) {
    // 아직 배송구역이 설정되지 않은 좌측 상단의 좌표를 구한다.
    var leastLeftY = -1;
    var leastLeftX = -1;
    for (var y = 0; y < area.length; y++) {
        for (var x = 0; x < area[0].length; x++) {
            if (area[y][x] === 0) {
                leastLeftY = y;
                leastLeftX = x;
                break;
            }
        }
        if (leastLeftY !== -1) break;
    }

    // 기저사례 : 모든 단지에 배송구역이 설정된 경우
    if (leastLeftY === -1) return 1;

    var result = 0;
    for (var typeIndex = 0; typeIndex < deliveryType.length;
typeIndex++) {
        // 배송구역을 설정한다.
        if (setDeliveryCase(area, leastLeftY, leastLeftX,
typeIndex, 1)) {
            // 설정한 배송구역이 유효하면 재귀호출 한다.
            result += countDeliveryCase(area);
        }
    }
}

```

```

    }

    // 설정한 배송 구역을 해제한다.
    setDeliveryCase(area, leastLeftY, leastLeftX, typeIndex,
-1);
    }

    return result;
}

// 배송구역을 나누는 종류, [dy, dx] 셋
var deliveryType = [
    [[0, 0], [0, 1], [0, 2]],
    [[0, 0], [1, 0], [2, 0]],
    [[0, 0], [0, 1], [1, 0]],
    [[0, 0], [0, 1], [1, 1]],
    [[0, 0], [1, 0], [1, 1]],
    [[0, 0], [1, 0], [1, -1]],
]

var setDeliveryCase = function (area, y, x, type, action) {
    var available = true;

    for (var i = 0; i < deliveryType[0].length; i++) {
        var newY = y + deliveryType[type][i][0];
        var newX = x + deliveryType[type][i][1];

        var isOutOfRange = newY < 0 || newY >= area.length ||
newX < 0 || newX >= area[0].length;
        if (isOutOfRange) {
            available = false;
        } else {
            var isDuplicated = (area[newY][newX] += action) > 1
            if (isDuplicated) {
                available = false;
            }
        }
    }

    return available;
}

```

테스트
케이스

예시케이스(1개)

입력			출력
h	w	areaText	
3	6	'@@...@@...@@@..@@'	2

정확성 테스트케이스(5개)

입력			출력
h	w	areaText	
3	3	'.....'	10

h w areaText

10 10 '@@@@@@@@@@@@@@@@@...@@@@@@@@@ 773
...@@@@@@@@@...@@@@@@@@@...@@@@@@@@@
...@@@@@@@@@...@@@@@@@@@...@@@@@...@
@@@@@@@@@@@@@@@@@'

출력