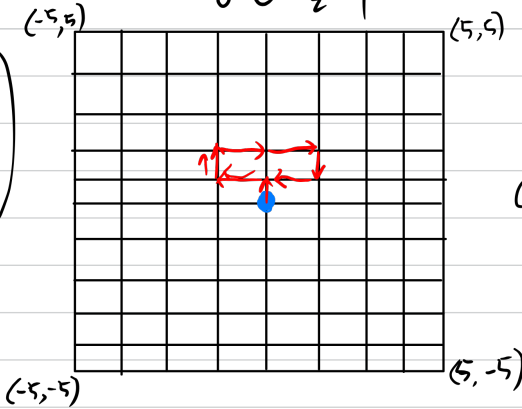
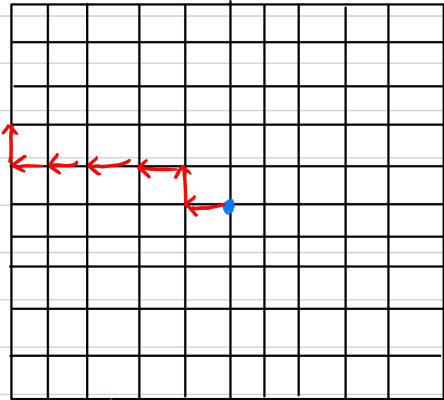


방문 길이

U: 위로
D: 아래로
R: 오른쪽으로
L: 왼쪽으로



(0,0)에서 시작
e.g., ULURRDLU
9번 움직였지만 처음 방문한: 7
(0,0) → (0,1) → (1,1) → (1,2) → (0,2)
→ (1,2) → (1,1) → (0,1) → (-1,1)
→ (-1,2)



e.g., LULLLLLLLU
(0,0) → (0,-1) → (1,-1) → (1,-2) → (1,-3)
→ (1,-4) → (1,-5) → (1,-6) → (1,-7) → (1,-8) → (1,-9) → (1,-10) → (2,-10)

Map key value, int[]

U	[0, 1]
D	[0, -1]
R	[1, 0]
L	[-1, 0]

→ 현재 좌표에서
해당 값은 4배됨.

초기 값 x, y → 0으로 지정
움직인 값 x2, y2 → x1 + int[0], y1 + int[1]

(1,0) → (2,1) 1,1 → 2,1) 등 4배됨. 등 4배한 마지막에 / 2 해줘야 함.

1. 이동 좌표랑 x, y 값 초기 세팅

```
Map<Character, int[]> location = new HashMap<>();
```

```
location.put('U', {0, 1});
```

```
location.put('D', {0, -1});
```

```
location.put('R', {1, 0});
```

```
location.put('L', {-1, 0});
```

```
int x = 0;
```

```
int y = 0;
```

2. 움직인 경로를 담은 HashSet 선언

```
Set<String> answer = new HashSet<>();
```

3. 경로 이동

```
for (int i = 0; i < dirs.length; i++)
```

```
    int[] loc = location.get(dirs.charAt(i));
```

```
    int x2 = x1 + loc[0];
```

```
    int y2 = y1 + loc[1];
```

3-1. -5, 5 넘어가면 무시해야 함.

```
if (x2 < -5 || x2 > 5 || y2 > 5 || y2 < -5) continue;
```

3-2. 안넘어가면 answer 에 경로 넣기

```
answer.add("(" + x1 + ", " + y1 + ") -> (" + x2 + ", " + y2 + ")");
```

```
answer.add("(" + x2 + ", " + y2 + ") -> (" + x1 + ", " + y1 + ")");
```

3-3. x1, y1 업데이트

```
x1 = x2;
```

```
y1 = y2;
```

4. 두 번씩 넣었으니 size() / 2 해서 리턴

```
return answer.size() / 2;
```

0,0 → 0,1
0,1 → 0,0
같은 경로