

---

# 자료구조 (특별)

## Chap 03. 스택 1 (미로 찾기)

2019년 하계 특별 학기

컴퓨터과학과  
민경하

---

# Contents

---

- 0. [7/11] STL
  - 1. [7/12] 배열 (sparse matrix)
  - 2. [7/15] 연결 리스트 (다항식)
  - 3. [7/16] 스택 1 (미로 찾기)
  - 4. [7/17] 스택 2 (수식 연산)
  - 5. [7/18] 트리 (조선 왕조)
  - 6. [7/19] 프로젝트 제안 발표
  - 7. [7/22] 탐색 트리 (AVL 트리)
  - 8. [7/23] 트리 & 힙 (최다 단어 찾기)
  - 9. [7/24] 그래프1 (이중 연결 요소)
  - 10. [7/25] 그래프2 (최단 거리 찾기)
  - 11. [7/25] 프로젝트 최종 발표
-

# 스택 1 (미로 찾기)

---

3.1 시작: Daedalos maze

3.2 미로의 표현

3.3 미로의 예제

3.4 미로의 자료 구조

3.5 미로 찾기 알고리즘

---

## 3.1 Daedalos maze

---

아테네는 크레타와의 전쟁에서 패해서 크레타의 왕 미노스에게 조공을 바치고 있었다. 조공은 일곱 명의 소년과 소녀로서 괴물의 밥이 되기 위해 매년 보내지는 것이었다.

그 괴물은 소의 몸뚱이와 인간의 머리를 가진 미노타우로스라는 짐승으로 다이달로스라는 사람이 만든 미궁 속에 갇혀 있었는데, 그 구조가 대단히 교묘하여 그 속에 갇힌 자는 누구나 혼자 힘으로 탈출하지 못하게 되어 있었다.

아버지의 뒤를 이어 후계자가 된 테세우스는 국민들의 고통을 덜어주기 위해 조공할 시기가 다가오자 자진해서 희생될 한 사람으로 나섰다. 그들은 크레타섬에 도착하여 미노스 왕앞에 나아갔다.

## 3.1 Daedalos maze

---

미노스왕의 딸 아리아드네는 테세우스의 모습을 보자 그를 사랑하게 되었다. 아프로디테 여신은 아리아드네가 테세우스를 사랑하도록 만들어 테세우스를 도와준 것이었다. 아리아드네는 다이달로스를 설득해 미궁을 빠져나올 방법을 알아냈다. 그녀는 테세우스에게 결혼을 약속받은 뒤, 테세우스에게 괴물을 썰기 위해 칼과 실 한 타래를 주고 입구에 실을 매고 이것을 따라 되돌아 나오는 방법을 알려주었다.

테세우스는 괴물을 처치하고 미궁으로부터 탈출하여 아리아드네를 동반하고 사람들을 구출해서 아테네를 향해 출발했다. 도중에 그는 낙소스 섬에 머물렀는데, 테세우스는 잠든 아리아드네를 그곳에 버리고 떠난다. 그가 은인에게 이와 같은 배은망덕한 짓을 한 것은 꿈에 아테나가 나타나 그렇게 하라고 명령했기 때문이었다. 후에 아리아드네는 디오니소스에게 발견되어 그의 아내가 된다.

## 3.1 Daedalos maze

---

미노스왕의 딸 아리아드네는 테세우스의 모습을 보자 그를 사랑하게 되었다. 아프로디테여신은 아리아드네가 테세우스를 사랑하도록 만들어 테세우스를 도와준 것이었다. 아리아드네는 다이달로스를 설득해 미궁을 빠져나올 방법을 알아냈다. **그녀는 테세우스에게 결혼을 약속받은 뒤, 테세우스에게 괴물을 썰어 칼과 실 한 타래를 주고 입구에 실을 매고 이것을 따라 되돌아 나오는 방법을 알려주었다.**

테세우스는 괴물을 처치하고 미궁으로부터 탈출하여 아리아드네를 동반하고 사람들을 구출해서 아테네를 향해 출발했다. 도중에 그는 낙소스 섬에 머물렀는데, 테세우스는 잠든 아리아드네를 그곳에 버리고 떠난다. 그가 은인에게 이와 같은 배은망덕한 짓을 한 것은 꿈에 아테나가 나타나 그렇게 하라고 명령했기 때문이었다. 후에 아리아드네는 디오니소스에게 발견되어 그의 아내가 된다.

## 3.1 Daedalos maze

---



## 3.1 Daedalos maze

---





## 3.1 Daedalos maze

---



## 3.1 Daedalos maze



## 3.1 Daedalos maze

---





## 3.1 Daedalos maze

---



## 3.1 Daedalos maze

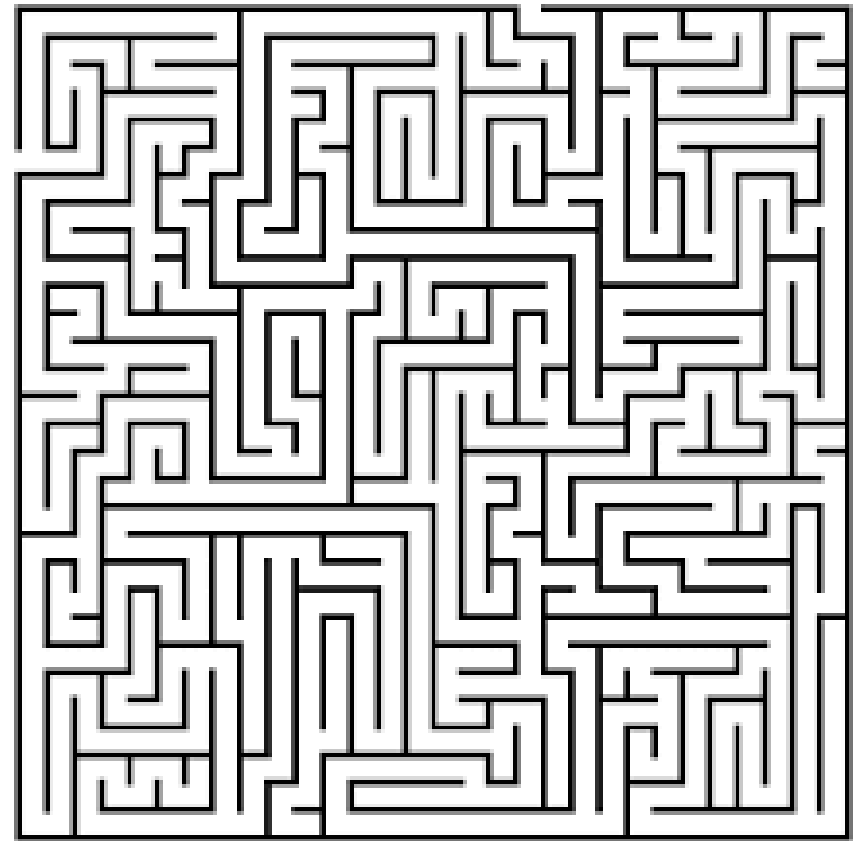
---





## 3.2 미로의 표현

- 미로?



## 3.2 미로의 표현

---

- 미로의 표현
  - 미로 = 길 (road) + 벽 (wall)
    - 벽 (통과할 수 없음)  $\rightarrow 1$
    - 길 (통과할 수 있음)  $\rightarrow 0$
  - 미로
    - 0과 1로 구성된 2차원 배열

```
111011
111001
101101
100001
110111
```

### 3.3 미로의 예제

- 예제 미로

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

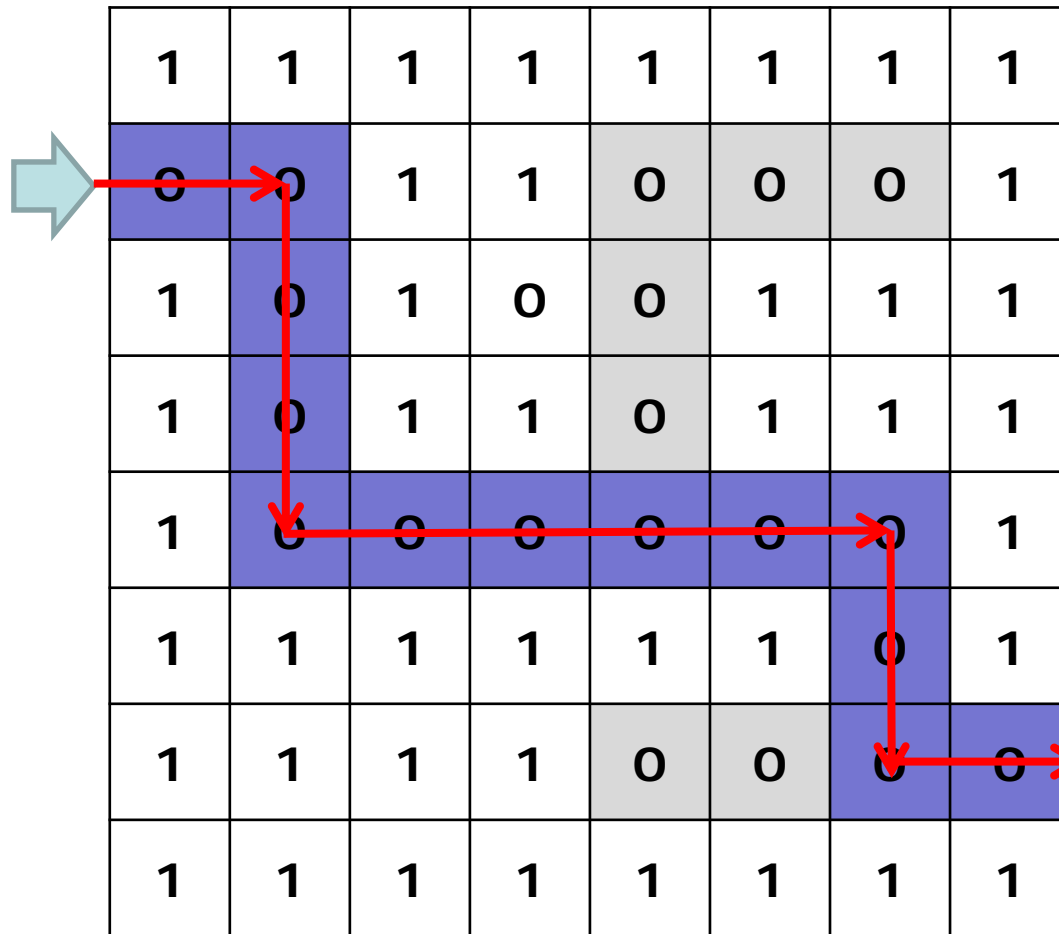
Entry: (1, 0)  
Exit: (6, 7)

- cell의 좌표: (x, y)
  - 원점: upper-left (0, 0)
  - X-axis: down
  - Y-axis: right



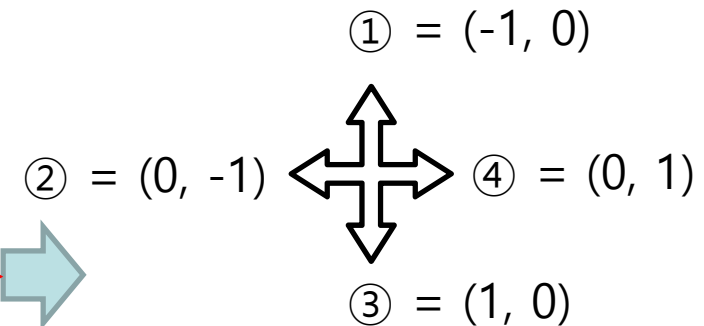
### 3.3 미로의 예제

- 예제 미로 → 경로 찾기



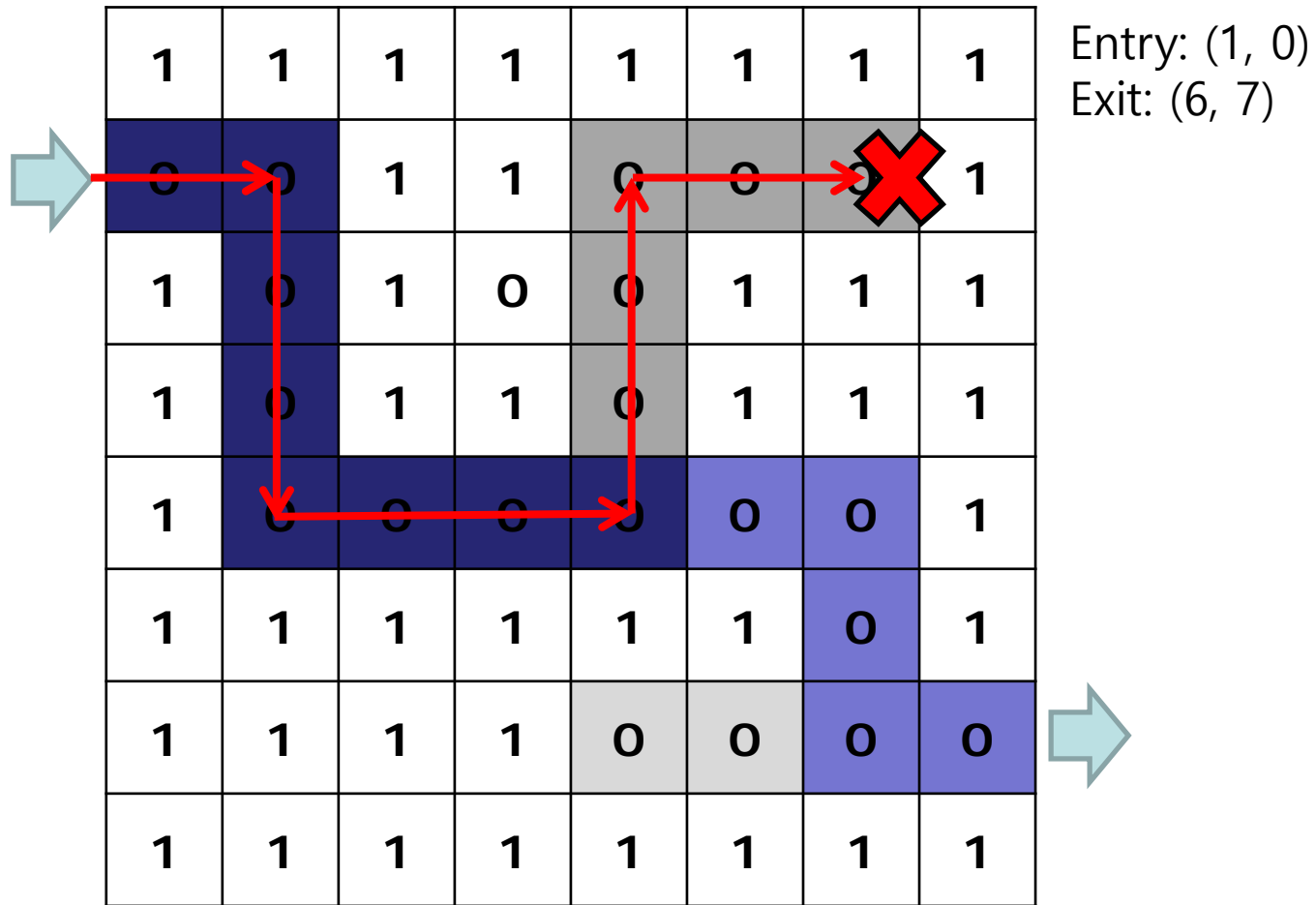
Entry: (1, 0)  
Exit: (6, 7)

- 4-nbhd or 8-nbhd
  - 4-nbhd를 사용
  - 방문 순서: ① ② ③ ④



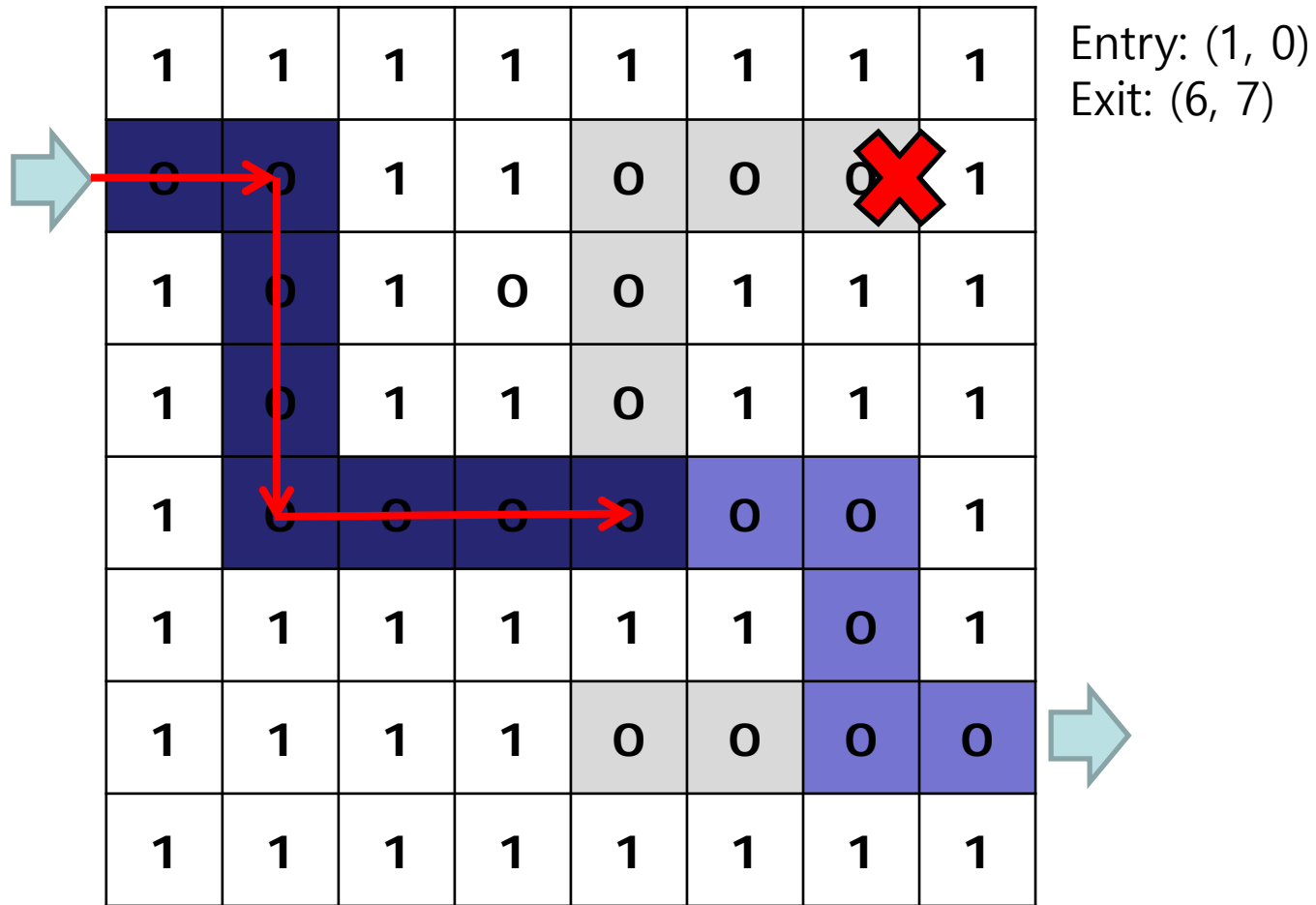
### 3.3 미로의 예제

#### (1) 갈수 있는 만큼 이동함



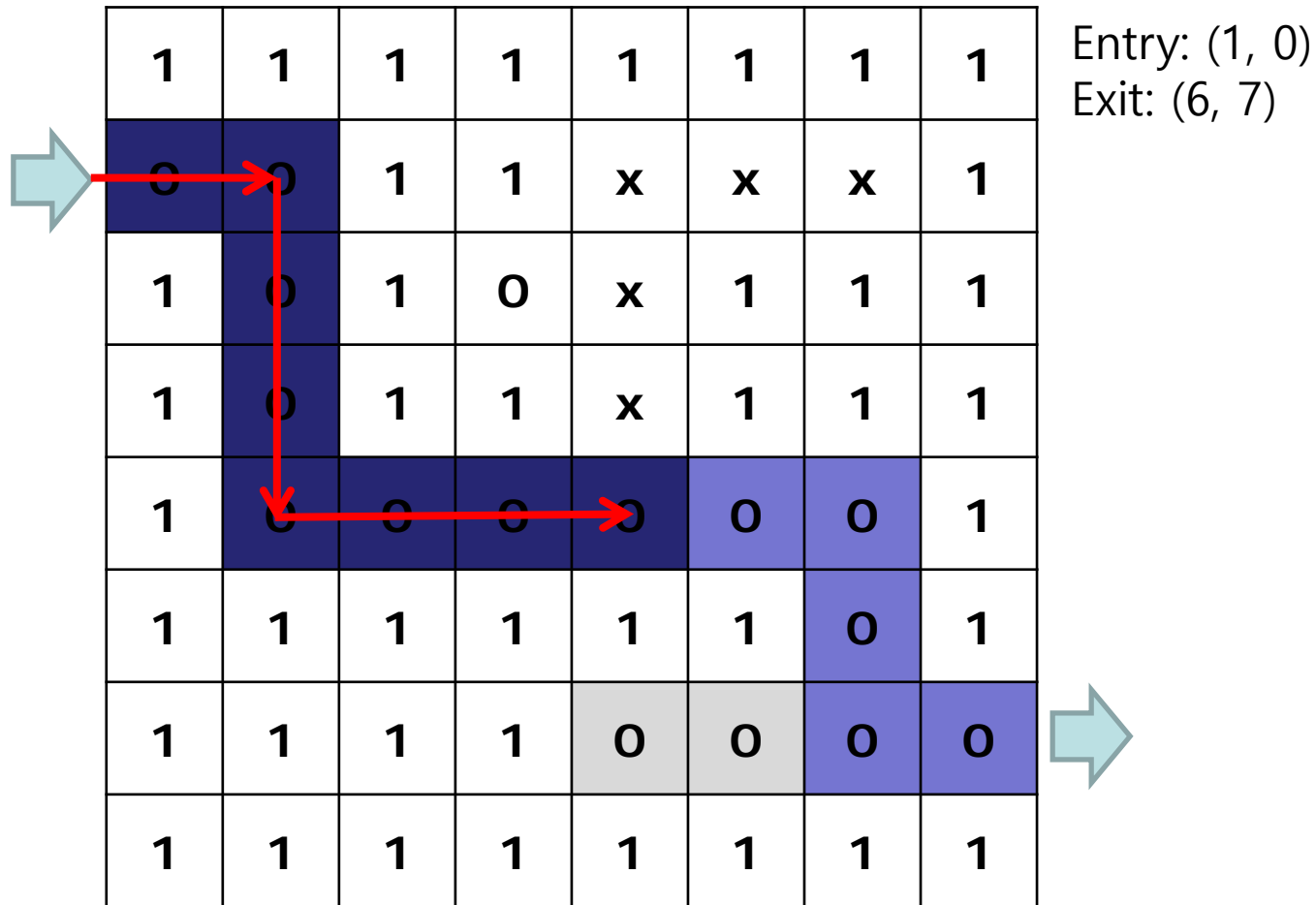
### 3.3 미로의 예제

(2) 더 이상 갈 수 없으면 가장 가까운 분기점으로 이동함



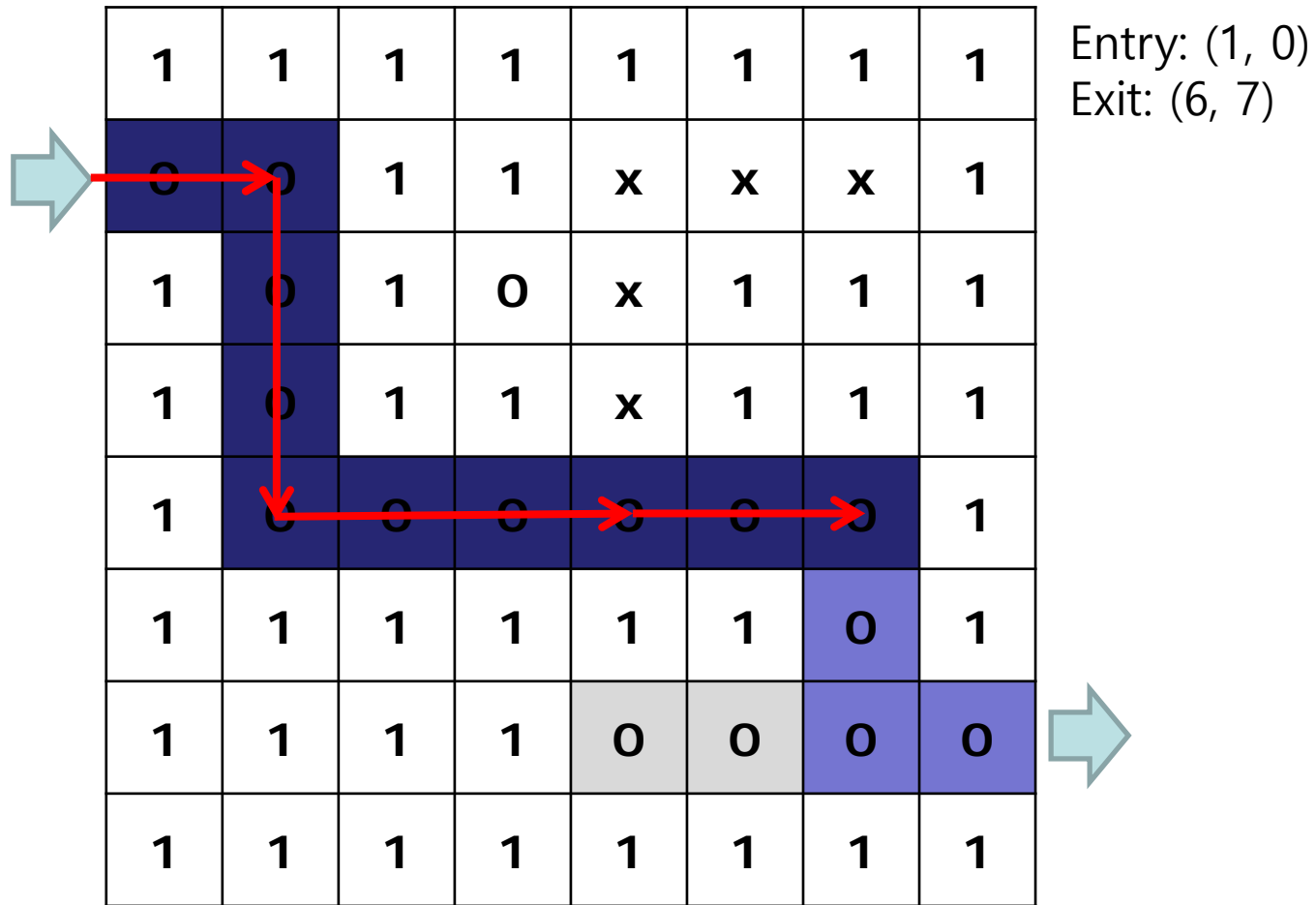
### 3.3 미로의 예제

(3) 되돌아온 길을 갈 수 없는 길로 표시



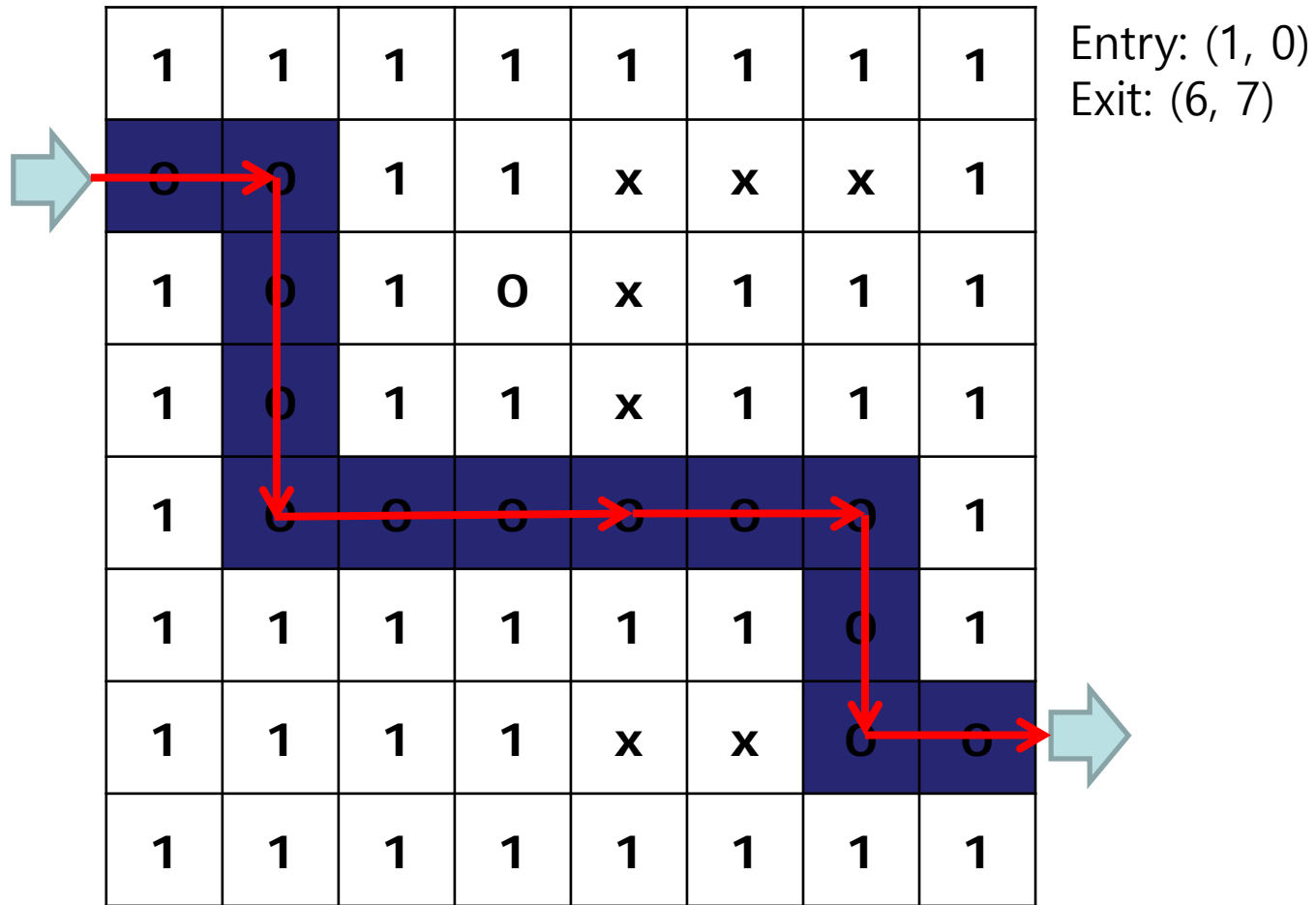
### 3.3 미로의 예제

(4) 분기점에서 다시 탐색을 시작함



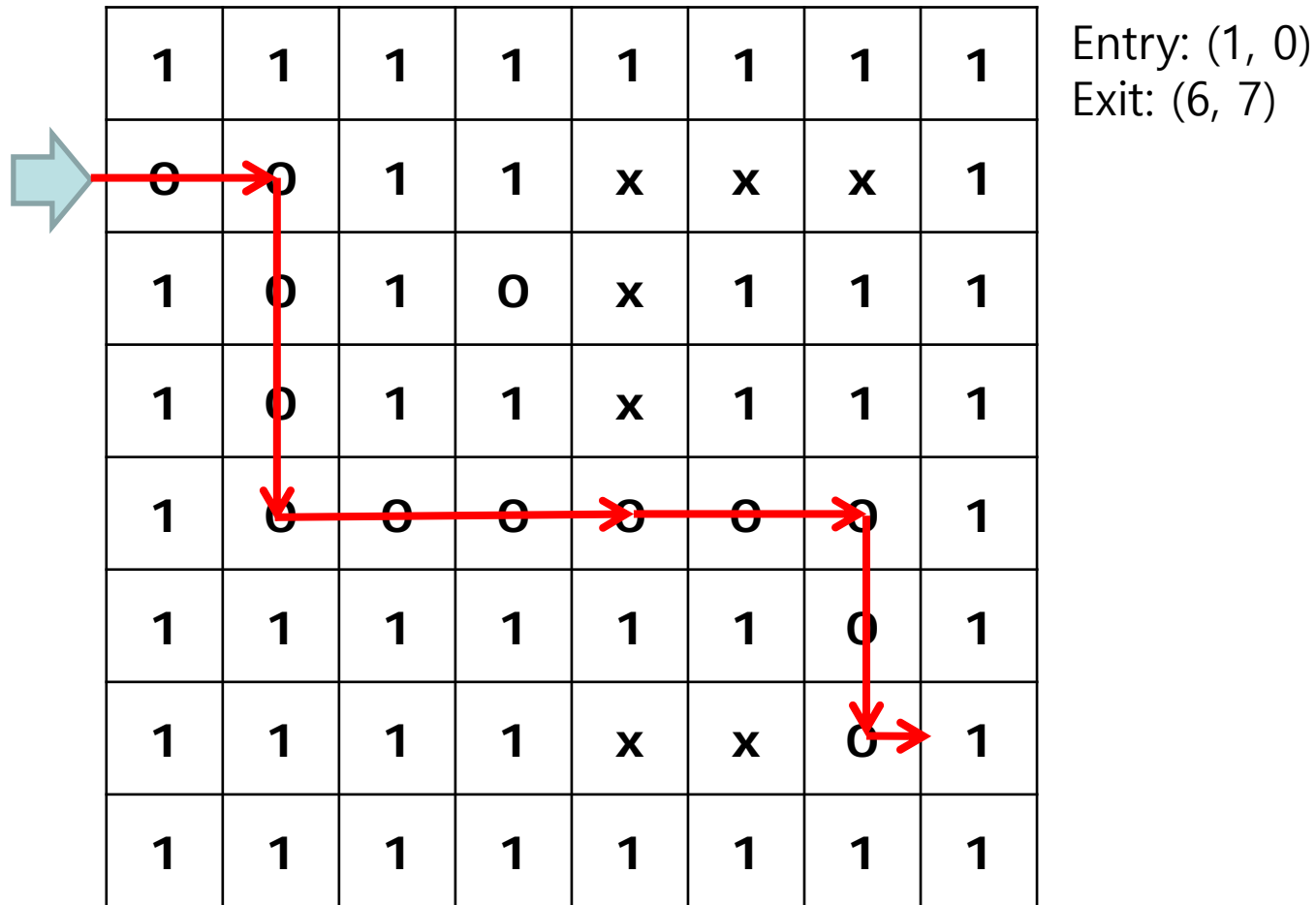
### 3.3 미로의 예제

(5) exit을 만나면 미로 찾기를 끝냄



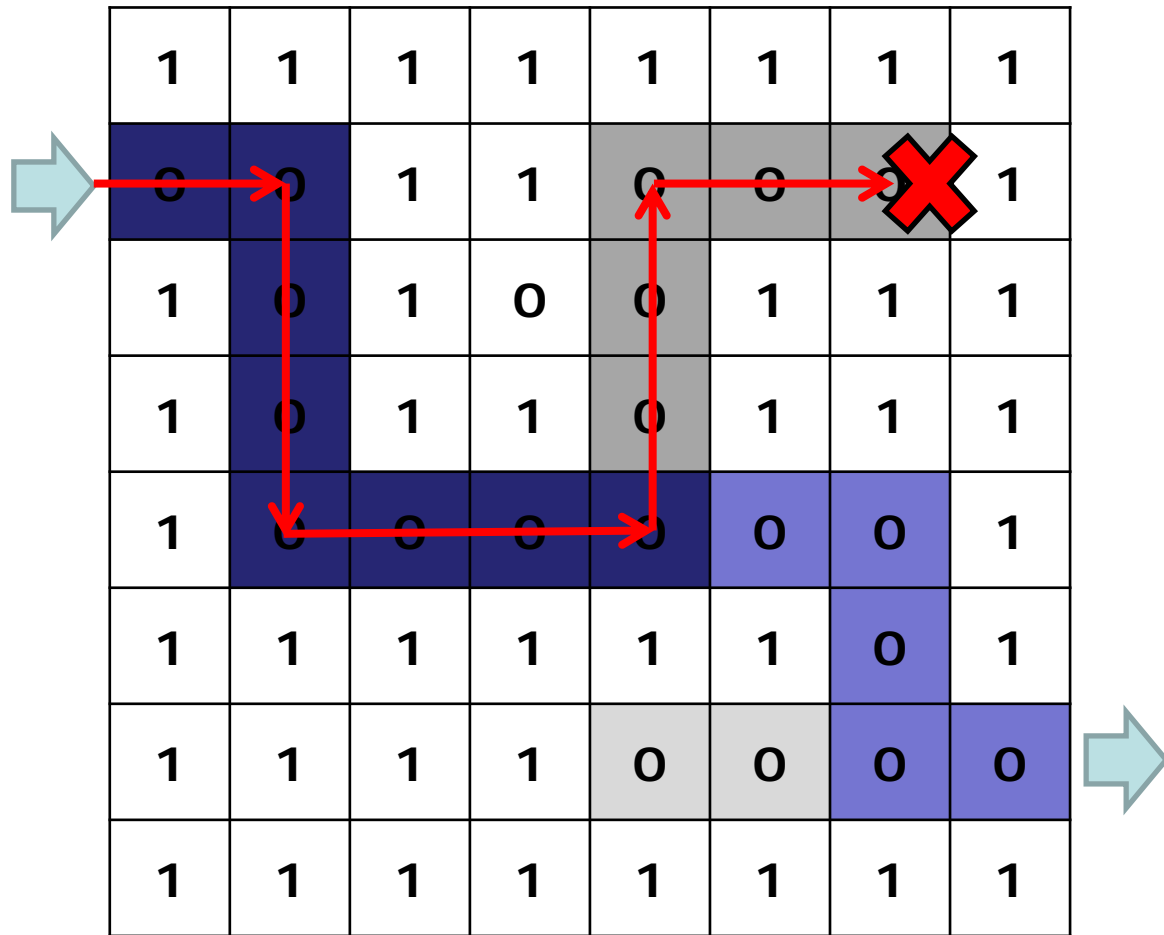
### 3.3 미로의 예제

(6) 다시 entry로 돌아오면 → 출구 없는 미로



### 3.4 미로의 자료 구조

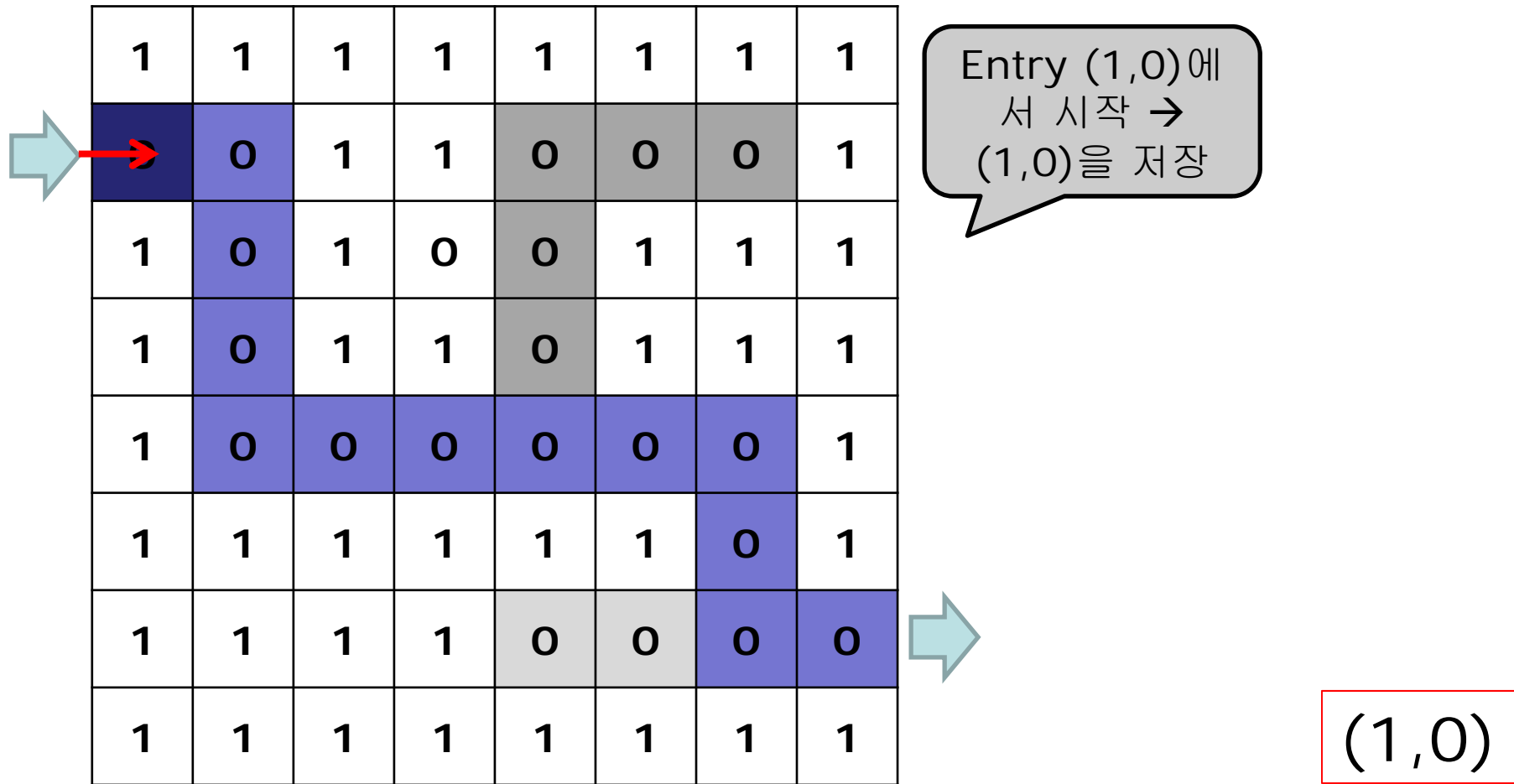
- 자료 구조: 실타래





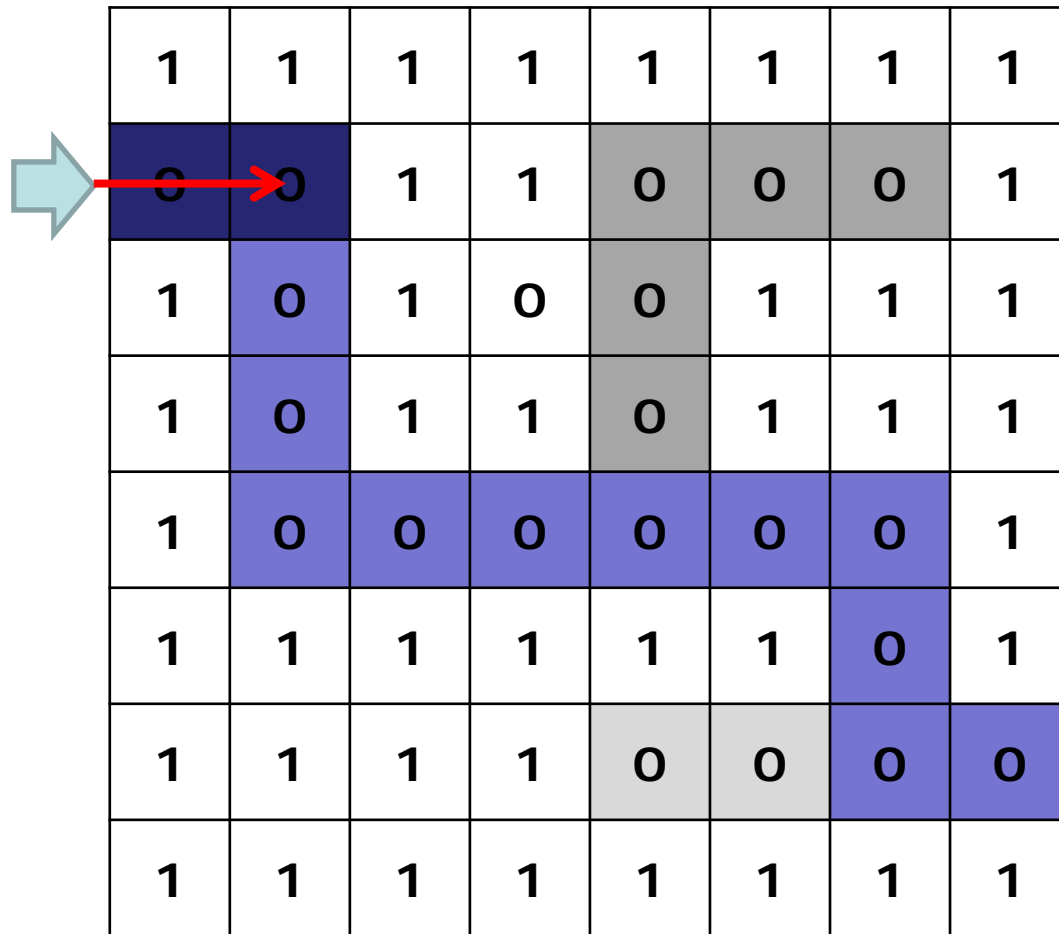
### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

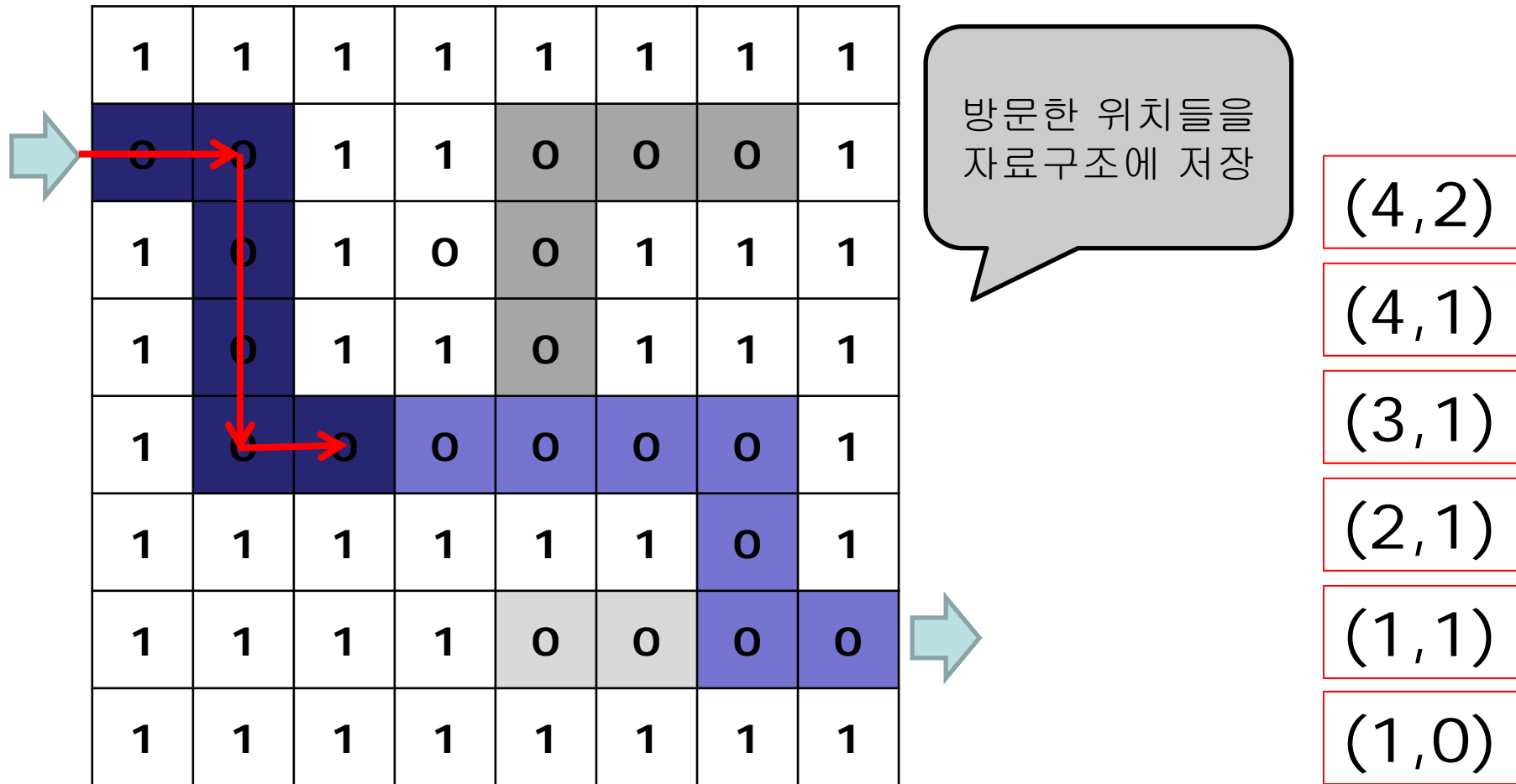
(1, 0)에서 다음  
위치를 결정함:  
(-1, 0), (0, -1),  
(1, 0), (0, 1)의  
순서로 탐색

(1, 1)

(1, 0)

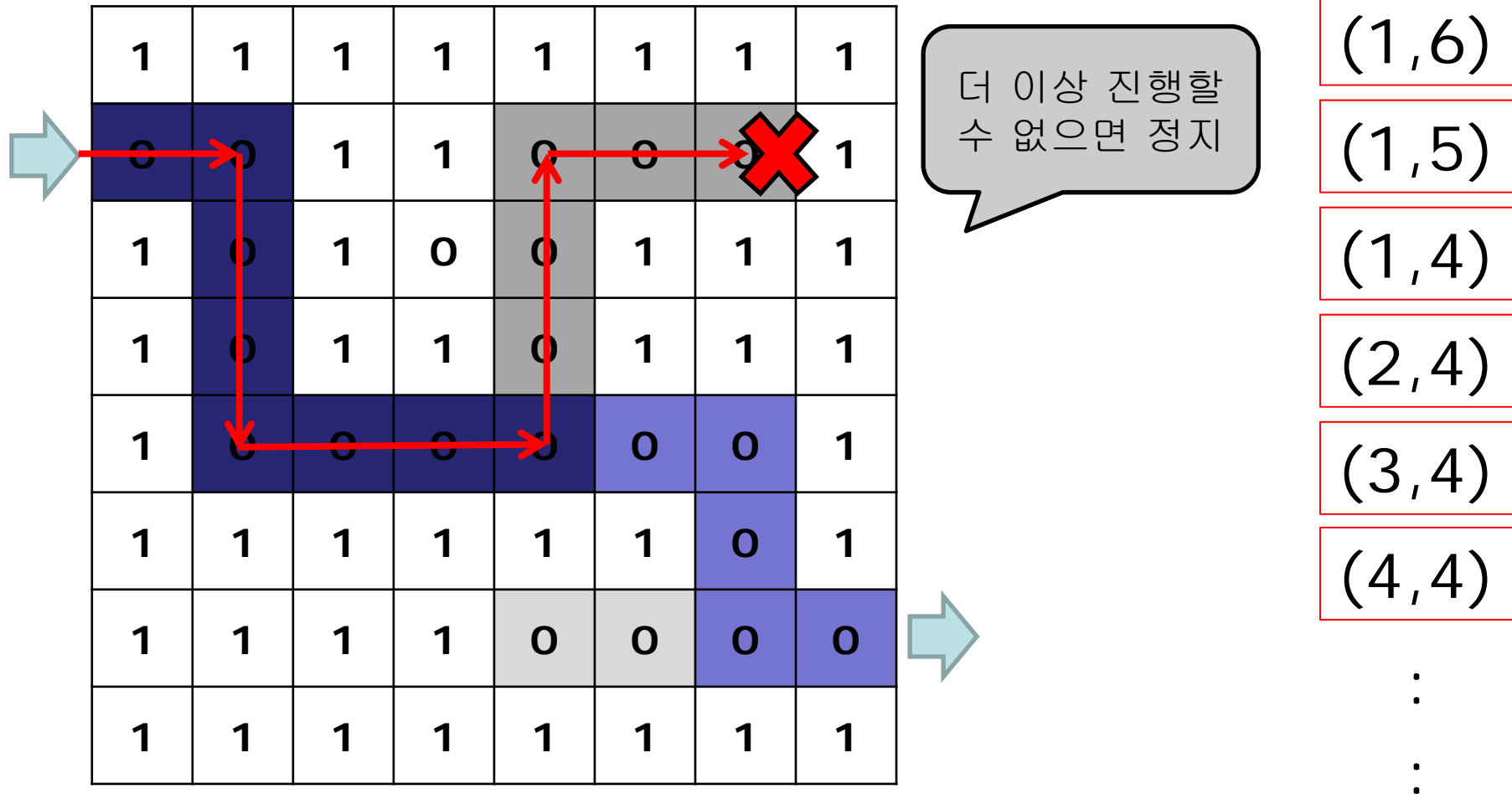
### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



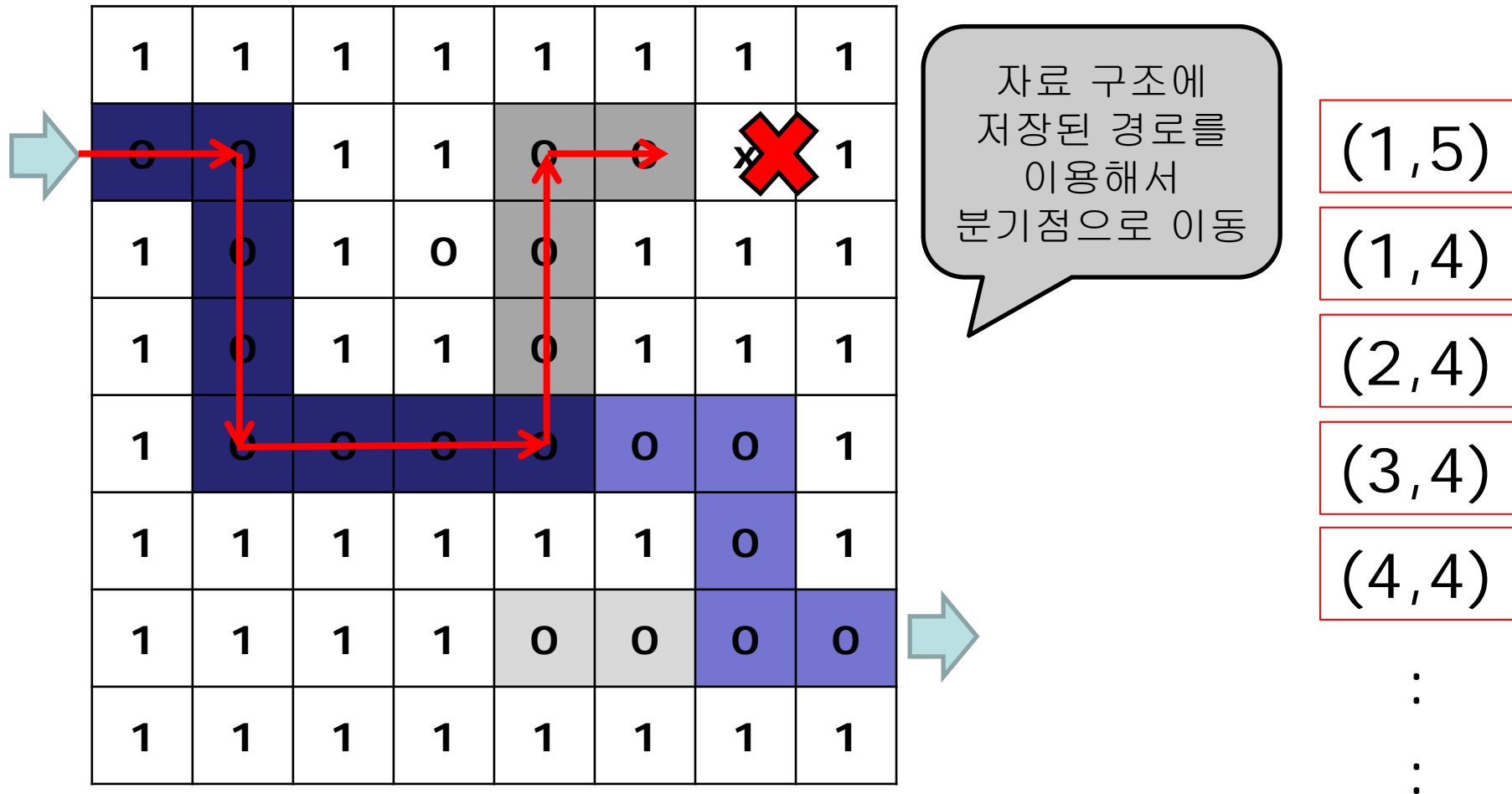
### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



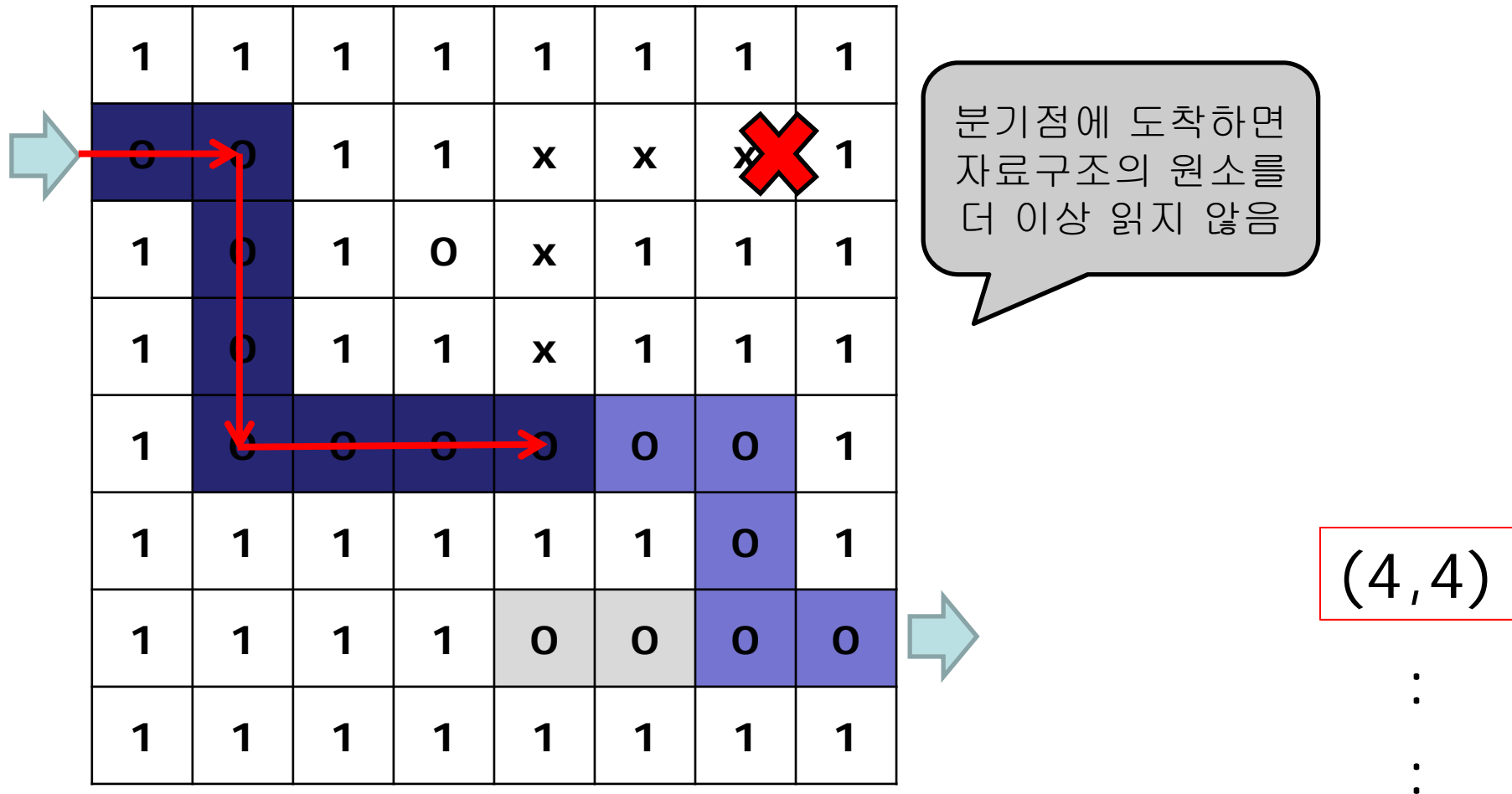
### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



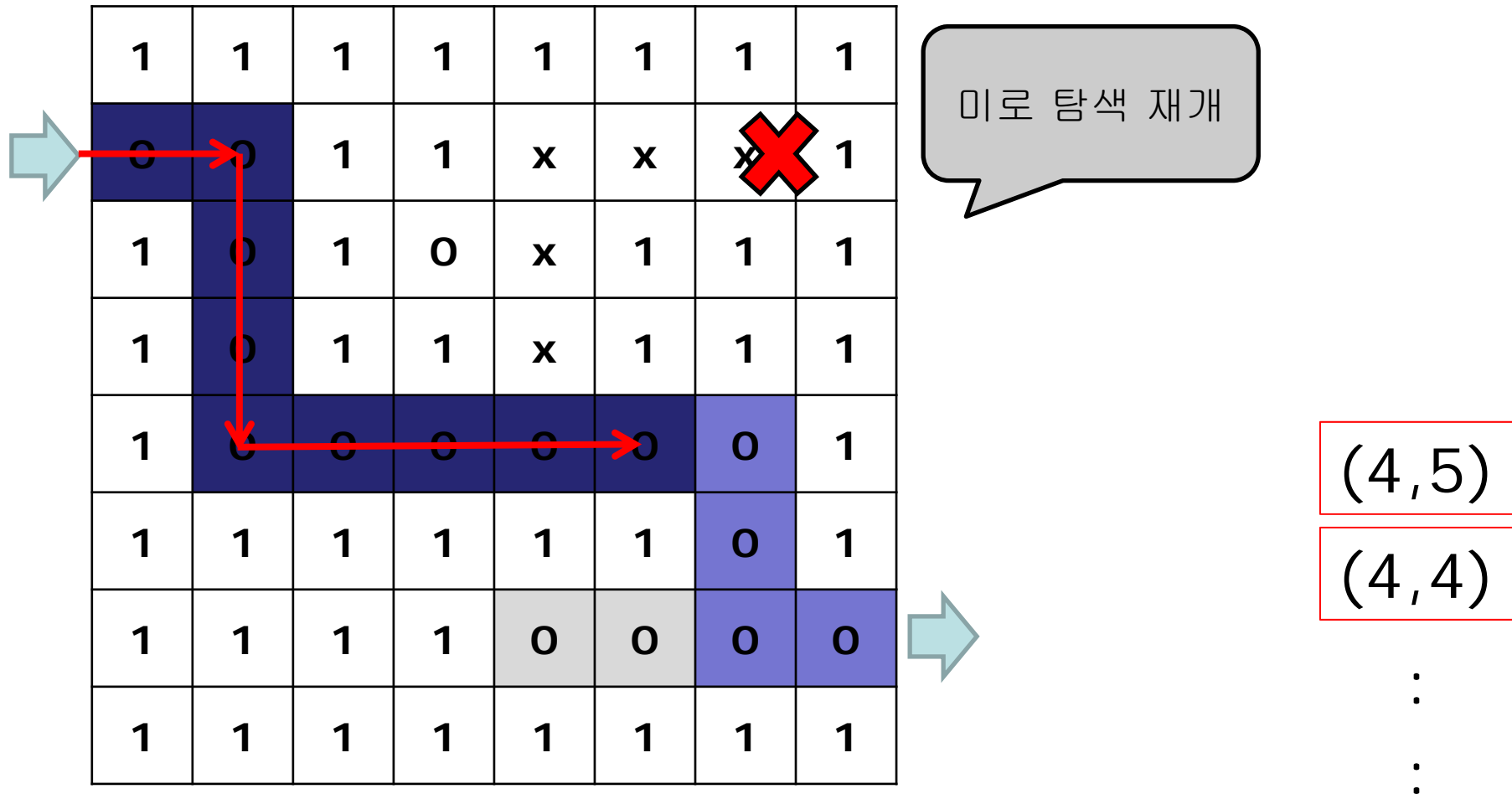
### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



### 3.4 미로의 자료 구조

- 자료 구조의 역할: 경로를 기록



## 3.4 미로의 자료 구조

---

- 자료 구조
  - 스택
  - 스택의 원소

```
class pos {  
    int x;  
    int y;  
};
```

```
class stack {  
    int top;  
    int n;  
    pos *arr;  
}
```



## 3.5 미로 찾기 알고리즘

---

- 스택을 이용한 미로 찾기 알고리즘
  - 입력: 미로, 입구, 출구
  - 출력: 입구에서 출구를 연결하는 경로

## 3.5 미로 찾기 알고리즘

---

- 스택을 이용한 미로 찾기 알고리즘
  - 이 알고리즘의 종결 조건은?
    - 출구를 찾았을 때
      - curr가 exit에 도착했을 때
    - 출구가 없음을 깨달았을 때
      - stack이 empty가 되었을 때

## 3.5 미로 찾기 알고리즘

```
int wid, hei;
pos in, out;
char **maze;
void solve_maze ( )
{
    // 1. 미로 찾기
    while ( !stack.is_empty ( ) ) {
        // 1. 이동할 수 있는 동안 이동
        //     이동할 때마다 위치를 stack에 저장
        //     출구에 도착하면 break

        // 2. 더 이상 진행할 수 없으면 분기점으로 되돌아가기
        //     이동할 수 있는 위치에 올 때까지 stack pop
    }

    // 2. 미로 출력: stack에 저장된 위치를 출력
}
```

# 프로그램 구성

---

- stack.h
- stack.cpp
- maze.cpp

# main ( ) 함수

---

```
int main()  
{  
    FILE *fp = fopen("maze.txt", "r+t");  
  
    load_maze(fp);  
  
    print_maze();  
  
    solve_maze();  
  
    print_maze();  
  
    fclose(fp);  
    system("pause");  
    return 0;  
}
```

---

# 입력

```
maze - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
10 10
1 1 1 1 1 1 1 1 1 1
E 0 0 0 0 0 0 0 0 0 1
1 1 0 1 1 1 0 1 0 1
1 1 0 1 0 0 0 0 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 1 1 1 0 1
1 1 0 0 0 0 0 1 1 1
1 1 1 1 1 1 1 1 0 X
1 1 1 1 1 1 1 1 1 1
```

```
maze2 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
10 10
1 1 1 1 1 1 1 1 1 1
E 0 0 0 0 1 0 0 1 1
1 1 0 1 1 1 1 0 1 1
1 1 0 1 0 0 0 0 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 1 1 1 0 1
1 1 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 0 X
1 1 1 1 1 1 1 1 1 1
```

C:\WHomeSSD\상명\2019년\특별 학기\W03 스택 1 (미로 찾기)\W실습\Wmaze... — □ ×

```
10, 10
1 1 1 1 1 1 1 1 1 1
E 0 0 0 0 1 0 0 1 1
1 1 0 1 1 1 1 0 1 1
1 1 0 1 0 0 0 0 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 1 1 1 0 1
1 1 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 0 X
1 1 1 1 1 1 1 1 1 1
```

Entry: 1, 0

Exit: 8, 9

1

```
10, 10
1 1 1 1 1 1 1 1 1 1
# # # 0 0 1 0 0 1 1
1 1 # 1 1 1 1 0 1 1
1 1 # 1 0 0 0 0 0 1
1 1 # 1 1 0 1 1 0 1
1 1 # 1 1 0 1 1 0 1
1 1 # 1 1 1 1 1 0 1
1 1 # # # # # # 1
1 1 1 1 1 1 1 # X
1 1 1 1 1 1 1 1 1 1
```

Entry: 1, 0

Exit: 8, 9

계속하려면 아무 키나 누르십시오 . . .

C:\HomeSSD\상명\2019년\특별 학기\W03 스택 1 (미로 찾기)\실습\maze... — □ ×

```
10, 10
1 1 1 1 1 1 1 1 1
E 0 0 0 0 0 0 0 0 1
1 1 0 1 1 1 0 1 0 1
1 1 0 1 0 0 0 0 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 0 1 1 0 1
1 1 0 1 1 1 1 1 0 1
1 1 0 0 0 0 0 1 1 1
1 1 1 1 1 1 1 1 0 X
1 1 1 1 1 1 1 1 1 1
```

Entry: 1, 0

Exit: 8, 9

No exit

```
10, 10
1 1 1 1 1 1 1 1 1 1
x x x x x x x x x 1
1 1 x 1 1 1 x 1 x 1
1 1 x 1 x x x x x 1
1 1 x 1 1 x 1 1 x 1
1 1 x 1 1 x 1 1 x 1
1 1 x 1 1 1 1 1 x 1
1 1 x x x x x 1 1 1
1 1 1 1 1 1 1 1 0 X
1 1 1 1 1 1 1 1 1 1
```

Entry: 1, 0

Exit: 8, 9

계속하려면 아무 키나 누르십시오 . . .