

# 알고리즘 - DFS/BFS

# DFS/BFS

깊이우선탐색 / 너비우선탐색

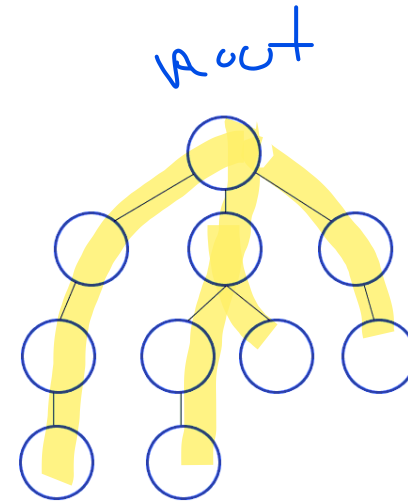
# DFS

## 깊이 우선 탐색 (Depth-First Search)

- Root 노드에서 다음 분기로 넘어가기 전에 해당 분기에 있는 모든 노드를 탐색하는 방법
- Stack 혹은 재귀함수로 구현

## 너비 우선 탐색 (Breadth-First Search)

- Root Node에서 인접한 노드를 먼저 탐색하는 방법
- Queue를 사용해서 구현



출처 : <https://developer-mac.tistory.com/64>

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

### 입력

- 첫 번째 줄에 얼음 틀의 세로 길이  $N$ 과 가로 길이  $M$ 이 주어진다. ( $1 \leq N, M \leq 1,000$ )
- 두 번째 줄부터  $N + 1$  번째 줄까지 얼음 틀의 형태가 주어진다.
- 이때 구멍이 뚫려있는 부분은 0, 그렇지 않은 부분은 1이다.

### 출력

- 한 번에 만들 수 있는 아이스크림의 개수를 출력한다.

### 입력

```
4 5
00110
00011
11111
00000
```

### 출력

```
3
```

---

```
00110
00011
11111
00000
```

---

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

어떻게 해결해야할까?

내가 물을 넣는다고 가정해보자.


0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

사람은 직관적으로 칸막이가 분리된 곳을 보고 확인이 가능하지만, 컴퓨터는 그것을 구분하기가 어렵다. 그래서, 한 칸 한 칸 규칙에 의해서 확인해나간다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0


## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

그러기 위해선 컴퓨터에게 다양한 것을 알려주어야 한다.

1. 아래와 같이 얼음틀에 대한 2차원 배열 정보
2. 해당 위치에 물이 담겨있는지 여부에 대한 정보
3. 확인해야 하는 위치를 담은 queue 혹은 stack과 같이 순서에 대한 정보

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

얼음 틀 2차원 배열				


물이 담았는지 확인할  
2차원 배열

[ (0, 0) ]

확인해야하는 위치 정보를  
담은 배열

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

현재 가지고 있는 정보를 가지고 컴퓨터처럼 하나씩 수행해보자.  
먼저, 확인해야할 목록 배열에서 0,0을 꺼낸다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0



[ (0, 0) ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

해당 위치가

- 1) 물을 담을 수 있는지
  - 2) 이미 물이 담겨있지는 않은지
- 확인한다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

(0, 0)				


[ ]



## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

둘 다 해당하지 않으므로, 물을 채운다. (색깔로 진하게 표시하는 걸로 구분한다.)

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

(0, 0)				


[ ]

예제 : 음료수 얼려먹기

- N × M 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의 4 × 5 얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

해당 위치는 물을 채웠으니, 현재 위치를 기준으로 상하좌우를 확인한다.

(-1, 0)

(0, -1)

(0, 0)	(0, 1)			
(1, 0)				


[ ]

예제 : 음료수 얼려먹기

- N × M 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의 4 × 5 얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

가장 먼저, 얼음틀을 벗어나는 위치는 제거한다.

(0, 0)	(0, 1)			
(1, 0)				


[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

그리고 나서, 바로 넣는 것이 아니라  
벽은 아닌지, 이미 물을 채운곳은 아닌지 확인하고 배열에 넣는다.

(0, 0)	(0, 1)			
(1, 0)				


[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

둘 다 해당사항 없으므로, 배열에 넣고 이번 회차를 종료한다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0



[ (0, 1), (1, 0) ]

예제 : 음료수 얼려먹기

- N × M 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의 4 × 5 얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

순서 목록에서 앞에서부터 꺼내거나 FIFO - queue  
뒤에서부터 꺼내서 확인한다. LIFO - stack

1	1	0	0	1
1	1	1	0	0
0	0	0	0	0
1	1	1	1	1

1	1	0	0	1
1	1	1	0	0
0	0	0	0	0
1	1	1	1	1

[ (0, 1), (1, 0) ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

앞에서 했던것처럼, 해당 위치가

- 1) 물을 담을 수 있는지
  - 2) 이미 물이 담겨있지는 않은지
- 확인한다.

	(0, 1)			


[ (1, 0) ]

예제 : 음료수 얼려먹기

- N × M 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의 4 × 5 얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

해당하지 않으면 물을 채우고, 그 다음 위치를 확인한다.

(0, 1)

(0, 0)	(0, 1)	(0, 1)		
	(0, 1)			


[ (1, 0) ]



## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

- 1) 틀을 벗어나지는 않는지
- 2) 벽이나 이미 물을 채운곳은 아닌지 확인했으면, 순서 목록에 넣는다.

	(0, 1)			
	(0, 1)			


[ (1, 0), (0, 1) ]

예제 : 음료수 얼려먹기

- N × M 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의 4 × 5 얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

이렇게 진행하다보면, 왼쪽 5칸의 얼음틀은 마무리가 되고, 순서 목록은 비게된다.



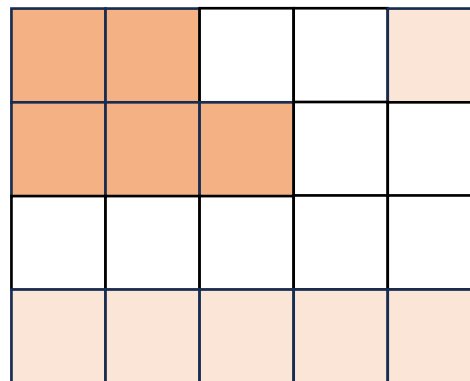
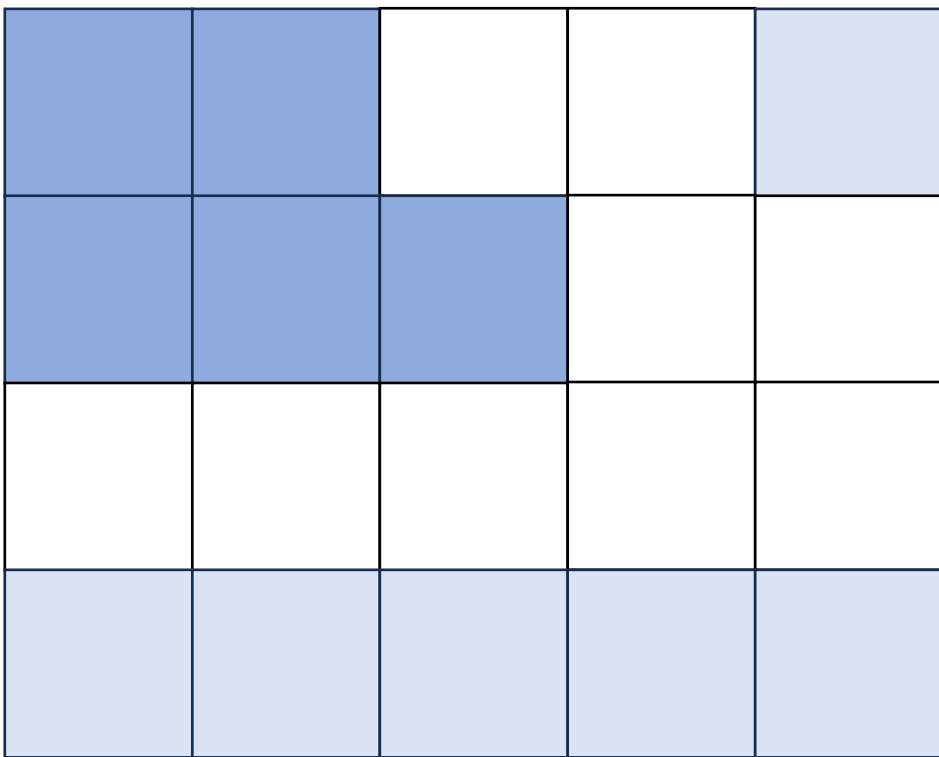
[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

문제를 다시 확인해보자. 문제에서 요구하는 것은 아이스크림을 총 몇 개를 만들 수 있는지다. 우리는 얼음을 만들다가 순서 목록이 비었을 때, 얼음틀에 빈 곳은 없는지 다시 확인하면 된다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0



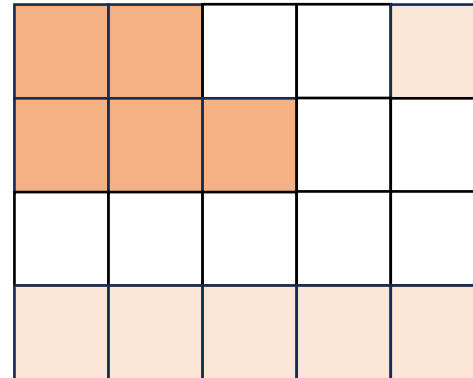
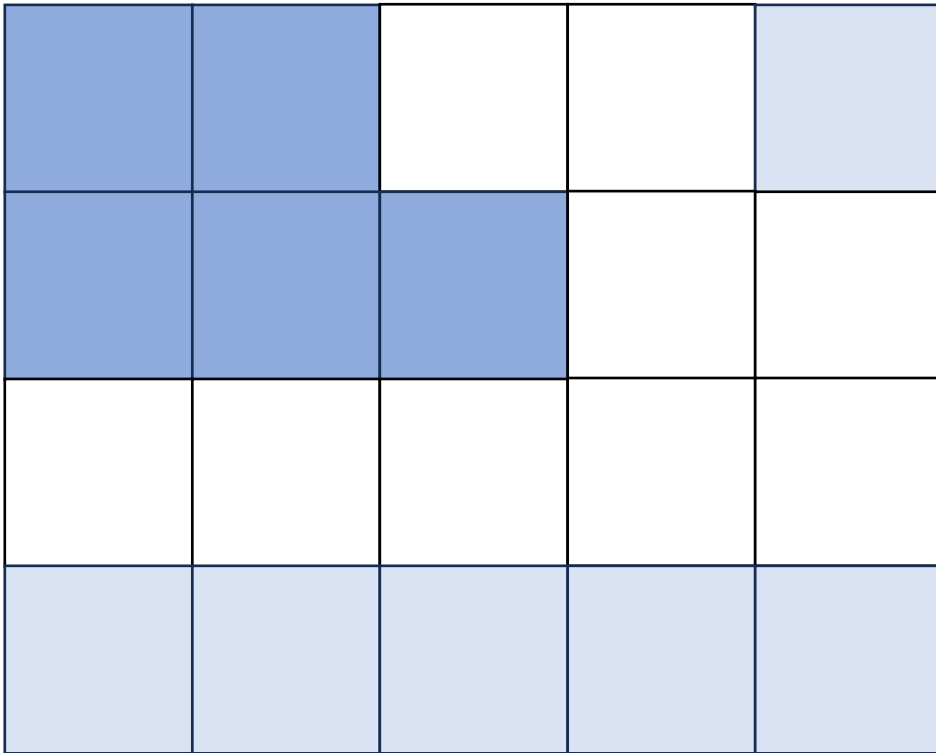
[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

(0, 0), (0,1), ..., (0, 4)! 위치를 보니, 아직 물을 넣지 않은 공간을 찾았다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0



[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

(0, 0), (0,1), ..., (0, 4)! 위치를 보니, 아직 물을 넣지 않은 공간을 찾았다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

				(0, 4)


[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.  
구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.  
이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.  
다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

이 위치를 순서 목록에 넣고, 앞의 과정을 반복한다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

				(0, 4)

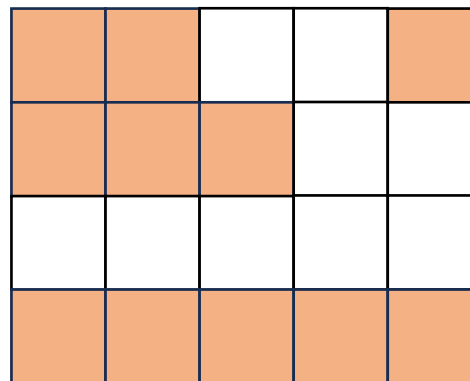
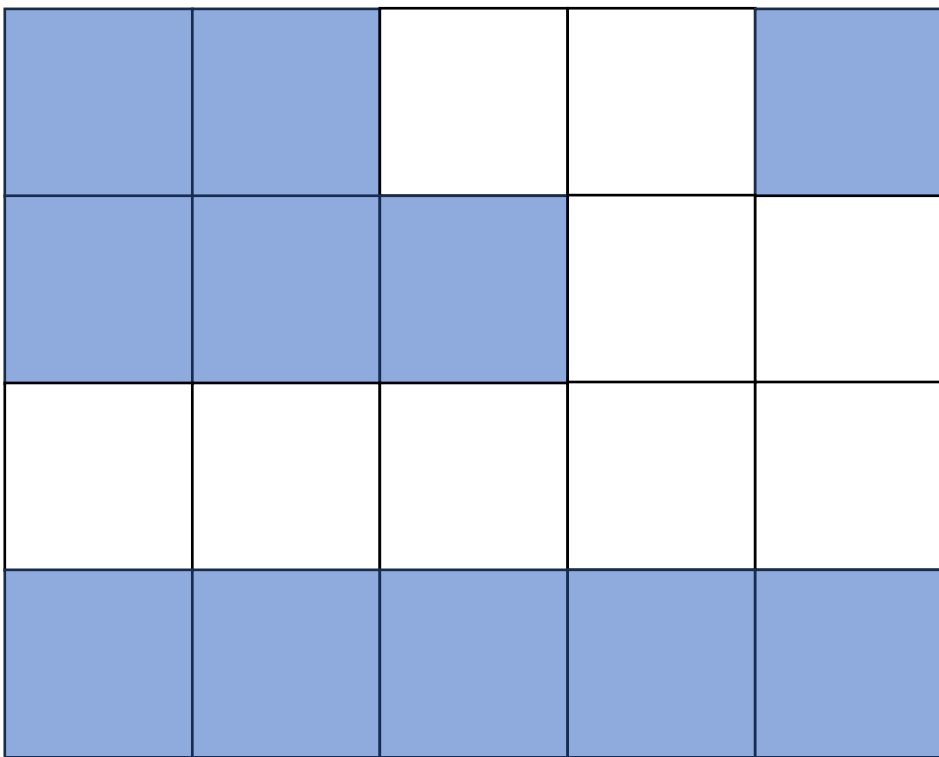

[ (0,4) ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

이렇게 진행하다보면, DFS/BFS를 수행한 횟수가 곧 아이스크림을 만들 수 있는 개수가 되고, 이를 반환하면 된다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0



[ ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

정리하면, 이 예제는

DFS/BFS를 수행한 횟수를 묻는 문제이다.

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0



## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

우리가 앞에서 수행한 것처럼

DFS/BFS는 다음의 과정으로 수행한다. (※ 수행에 필요한 정보는 미리 만들어 둔다.)

1) DFS/BFS를 수행할 시작 위치를 순서 목록에 담는다.

[ (0, 0) ]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

우리가 앞에서 수행한 것처럼

DFS/BFS는 다음의 과정으로 수행한다. (※ 수행에 필요한 정보는 미리 만들어 둔다.)

- 1) DFS/BFS를 수행할 시작 위치를 순서 목록에 담는다.
- 2) 해당 위치를 사용(방문)했음을 체크한다.

(0, 0)				

[]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

우리가 앞에서 수행한 것처럼

DFS/BFS는 다음의 과정으로 수행한다. (※ 수행에 필요한 정보는 미리 만들어 둔다.)

- 1) DFS/BFS를 수행할 시작 위치를 순서 목록에 담는다.
- 2) 해당 위치를 사용(방문)했음을 체크한다.
- 3) 갈 수 있는 방향을 통해 다음 위치를 확인한다.
  - 3-1) 틀을 벗어나지는 않는지
  - 3-2) 이미 사용하거나 방문하지는 않았는지

	(0, 1)			
(1, 0)				

[]

## 예제 : 음료수 얼려먹기

-  $N \times M$  크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라. 다음의  $4 \times 5$  얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

우리가 앞에서 수행한 것처럼

DFS/BFS는 다음의 과정으로 수행한다. (※ 수행에 필요한 정보는 미리 만들어 둔다.)

- 1) DFS/BFS를 수행할 시작 위치를 순서 목록에 담는다.
- 2) 해당 위치를 사용(방문)했음을 체크한다.
- 3) 갈 수 있는 방향을 통해 다음 위치를 확인한다.
  - 3-1) 틀을 벗어나지는 않는지
  - 3-2) 이미 사용하거나 방문하지는 않았는지
- 4) 갈 수 있는 위치를 목록에 넣고, 위의 과정을 반복한다.


[ (0, 1), (1, 0) ]

실전 : 게임 맵 최단 거리 (Lv.2)

<https://school.programmers.co.kr/learn/courses/30/lessons/1844>

실전 : 타겟 넘버 (Lv.2)

<https://school.programmers.co.kr/learn/courses/30/lessons/43165>

실전 : 네트워크 (Lv.3)

<https://school.programmers.co.kr/learn/courses/30/lessons/43162>

실전 : 단어변환(Lv.3)

<https://school.programmers.co.kr/learn/courses/30/lessons/43163>