

수강생ID	
성명	

Mission #6

(섬의 개수 구하기)

Introduction

정사각형으로 이루어져 있는 섬과 바다 지도가 주어진다. 섬의 개수를 세는 프로그램을 작성하시오.

Objectives

이번 Mission을 수행하면 다음과 같은 Skill들을 이해하고 적용할 수 있다.

- 배열을 이용한 너비우선탐색(BFS) 알고리즘을 구현할 수 있다.
- Flood Fill 기법을 이해하고 문제해결에 적용할 수 있다.

Requirement

정사각형으로 이루어져 있는 섬과 바다 지도가 아래 [그림 1] 과 같이 주어질 때 지도내의 섬의 개수가 총 몇 개인지 구하는 프로그램을 작성하시오.

각각의 정사각형이 하나의 섬으로 인식되려면, 한 정사각형에서 다른 정사각형으로 걸어서 갈 수 있는 경로가 있어야 한다. (하나의 정사각형 기준으로 8방향으로 연결될 수 있음 - 위, 아래, 왼쪽, 오른쪽, 대각선 4방향)

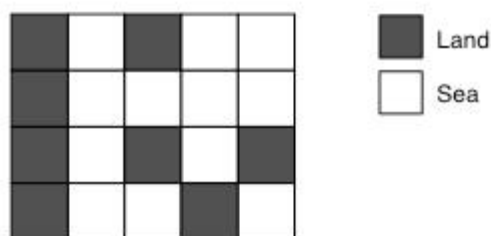
지도는 바다로 둘러 쌓여 있으며, 지도 밖으로 나갈 수 없다고 가정한다.

데이터파일은 여러 개의 테스트 케이스로 이루어져 있다.

각 테스트 케이스의 첫째 줄에는 지도의 너비 w 와 높이 h 가 주어진다. w 와 h 는 50보다 작거나 같은 양의 정수이다.

둘째 줄부터 h 개 줄에는 지도가 주어진다. 1은 땅, 0은 바다이다.

데이터파일로 부터 데이터 입력 시 입력 종료조건은 w, h 위치에 0 0이 입력되면 종료한다.



[그림 1] 섬의 개수 3개

Data Structure & Algorithm Mission #6

(Land.txt 데이터 파일의 구조)

1 1 ← Test Case1

0

2 2 ← Test Case2

0 1

1 0

3 2 ← Test Case3

1 1 1

1 1 1

5 4 ← Test Case4

1 0 1 0 0

1 0 0 0 0

1 0 1 0 1

1 0 0 1 0

5 4 ← Test Case5

1 1 1 0 1

1 0 1 0 1

1 0 1 0 1

1 0 1 1 1

5 5 ← Test Case6

1 0 1 0 1

0 0 0 0 0

1 0 1 0 1

0 0 0 0 0

1 0 1 0 1

50 50 ← Test Case7 (데이터의 내용이 너무 많아 생략 함. 제공된 파일 참고)

.....

.....

Data Structure & Algorithm Mission #6

provide file

- Data file : land.txt
- Header file : circular_queue.h
- Source file : circular_queue.cpp, island_count_bfs_file.cpp(미완성본)

Submit file

- 사번_AD계정ID_ island_count_bfs_file.cpp
※ source 파일만 제출하세요.

실행 예

```
0
섬의 개수 : 0개 ← Test Case1

0 1
1 0
섬의 개수 : 1개 ← Test Case2

1 1 1
1 1 1
섬의 개수 : 1개 ← Test Case3

1 0 1 0 0
1 0 0 0 0
1 0 1 0 1
1 0 0 1 0
섬의 개수 : 3개 ← Test Case4

1 1 1 0 1
1 0 1 0 1
1 0 1 0 1
1 0 1 1 1
섬의 개수 : 1개 ← Test Case5

1 0 1 0 1
0 0 0 0 0
1 0 1 0 1
0 0 0 0 0
1 0 1 0 1
섬의 개수 : 9개 ← Test Case6

.....

.....

섬의 개수 : 170개 ← Test Case7
```