

---

## PA 4 :

2 차원 배열에 빙산과 바다가 채워져 있고 빙산의 높이가 저장되어 있다. 높이는 일년 마다 그 칸에 동서남북 네 방향으로 붙어있는 0 이 저장된 칸의 개수만큼 줄어든다. 전체의 빙산이 두 덩어리가 될 때까지의 년도를 구하고 다 녹을 때 까지 분리되지 않으면 0 을 출력한다.

---

### Facts :

- 빙산과 바다의 정보를 저장할 배열
- DFS 도중 업데이트하면 결과값이 달라지니 임시 빙산의 정보를 저장할 배열 필요
- 빙산 노드의 정보를 담을 구조체
- 결과값

---

### Overview :

유저로부터 바다와 빙산의 정보를 입력 받아 구조체를 생성해 2차원 배열에 저장하고, 인접 리스트를 만들어 DFS 알고리즘을 통해 확인한다. 빙산의 경우 DFS를 했을 때 전체 빙산의 개수와 일치하지 않으면 아마도 빙산은 2개 이상일 것이다.

---

### Algorithm :

바다와 빙산의 정보인 2차원 배열에 있는 높이의 값이 0이 아닌 경우 : 상,하,좌,우를 검사해 0인 개수를 구하고 그만큼 뺀 값을 임시 바다와 빙산의 정보에 넣는다. -> 모든 빙산구조체에 대해 수행

위의 작업이 1년이 지나는 과정이다.

1년이 지나면 DFS를 통해 검사한다. 임의의 빙산에 대해 연결된 개수가 전체 빙산의 개수와 일치하지 않는 경우에는 2덩어리 이상일 것이다.

---

***Time Complexity :***