

백준 2573 - 빙산

입력: 세로크기 n , 가로크기 m $3 \leq n, m \leq 300$

이어서 빙산의 높이정보가 입력된다. $0 \leq h \leq 10$

빙산이 있는 칸의 개수는 항상 10000개 이하.

조건: 빙산이 분리되는 최초의 시간을 출력한다.

시간복잡도: 배열을 한 번씩 검사: $n \times m \leq 90000$

⊕ 빙산인 칸을 사방탐색: $10000 \times 4 \times 10 = 400000$
 ≈ 500000

시간제한 = 1초이때문에 충분함 OK

필요한 변수:

- `int curr` // 현재 탐색해본 빙산영역.
- `int tot` // 현재 빙산이있는 총 영역의 수를 저장
- `int[n][m] map` // 현재 빙산의 상태를 저장
- `boolean[n][m] visited` // 방문한 곳인지 여부를 체크할 배열
- `Queue<Point> q` // bfs 탐색을 위한 큐.
- `boolean isSeparated` // 빙산이 분리되었는지 체크할 변수
- `int year`

조직:

- (1) 입력정보를 map에 저장,
 입력받은 `tot` ++
 9이 좌표 저장

(2) 빙산탐색 조직

`while (!isSeparated) {`

`for (i: n) {`

`for (j: m) {`

`if (map[i][j] != 0 && !visited)`

`visited = true, year++,`

`q.add(현재좌표), size=q.size`

`while (!q.isEmpty) {`
`q.poll.`

// 4방향 탐색
 → 바다인 부분: 현재높이 감소
 → 빙산인부분: 큐에 넣는다. `curr++`.

`}`

// 탐색이 tot != curr이면 탐색종료, year 증가

year = 0일때까지 x면 0 출력.

*visited = true이면 무조건 방문.
 (false도 0일때만 방문.)*

구분방법.