

[22.03.25] 집 만들기

# TS집크기구하기

김 태 현



# 문 제

$N * N$  크기의 게임 판에 점을 놓는다.

점 주변 8방향에 점이 있는 경우 같은 그룹으로 표현

같은 그룹의 점들이 빈 공간들 주위를 둘러싸으면, 집이라고 하고 빈 공간의 개수가 집의 크기이다.

## 점이 추가되는 규칙

1. 빈 공간에만 추가
2. 모든  $2*2$  크기의 영역에 놓인 점들은 2개 이하
3. 점을 놓았을 때, 만들어지는 집은 하나
4. 이미 만들어진 집 내부에 점을 놓지는 않음
5. 집이 만들어질 때, 내부에 연결되지 않은 점/집 은 존재하지 않음

※ 점이 추가되는 횟수보다 점이 추가되어 집이 만들어지는 횟수가 상대적으로 적다.

따라서 점이 추가될 때, 집이 만들어지는 여부를 빠르게 알수 있다면 그 프로그램은 좋은 성능을 가질 수 있다.

점이 추가되었을 때 집이 만들어지는지 여부는 주변 점들의 그룹 정보를 이용하면 빠르게 확인할 수 있다.

```
void init(int N)
```

$N * N$  크기의 빈 게임 판 생성

```
int play(int row, int col)
```

(row, col) 에 점 추가

점을 놓았을 때 집이 만들어지면 그 집의 크기 반환

집이 만들어지지 않으면 0 반환

$5 \leq N \leq 120$

`play()`  $\leq 8,000$

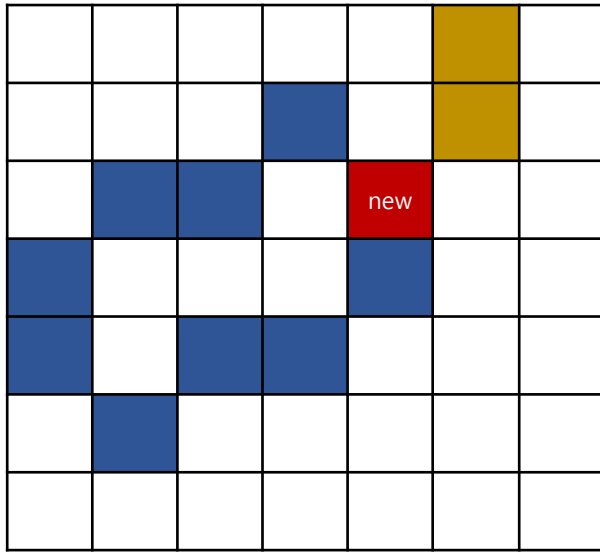
집이 만들어지는 횟수  $\leq 600$

# 주요 로직

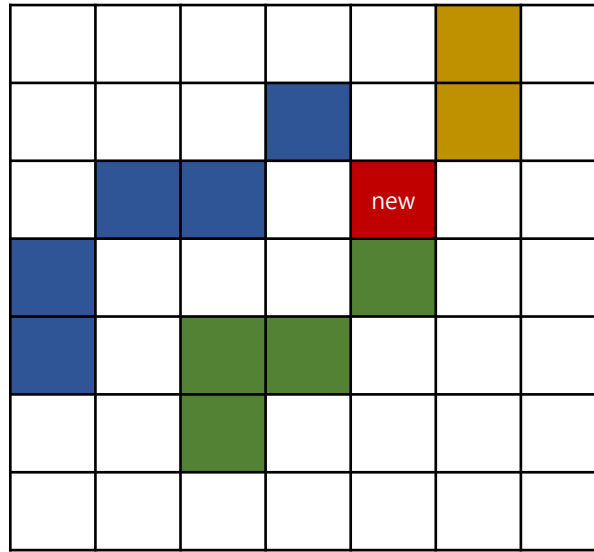
1. 집이 만들어지는지 판별
2. 놓은 점 주변을 같은 그룹으로 변경
3. 만들어진 집 크기 구하기

# 집이 만들어지는지 판별

집이 만들어지는 경우

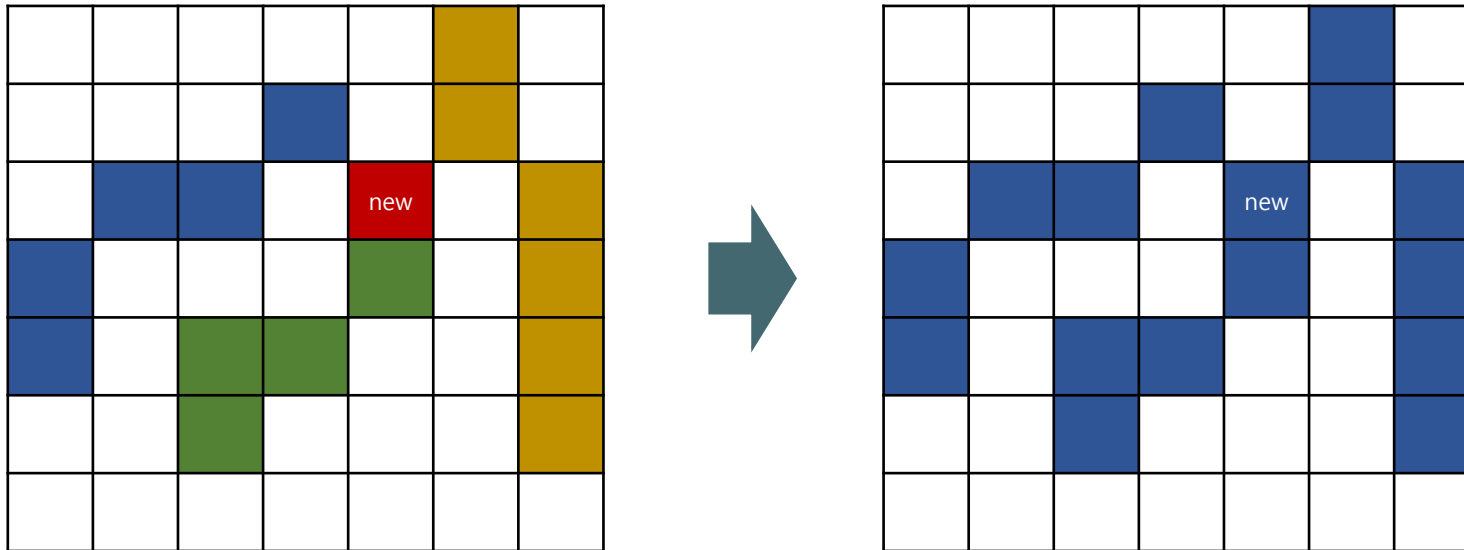


집이 만들어지지 않는 경우



새로운 점의 인접한 점들 중, 같은 그룹인 점이 존재하면 집이 만들어짐  
새롭게 만들어지는 집은 한개뿐이므로 같은 그룹인 점은 오로지 두개

# 놓은 점 주변을 같은 그룹으로 변경



기준을 한 개 잡아서 연결된 모든 점들을 탐색하며 변경  
=> dfs or bfs

# 만들어진 집 크기 구하기

2	2	2	2	2		3
2	2	2		2		3
2			1	new	3	3
	1	1	1		3	3
	1			3	3	3
3		3	3	3	3	3
3	3	3	3	3	3	3

2	2	2	2	2	2				
2	2	2	2	2	2				
2	2	2			2	2			
2			1	1		2	2		
	1	1	1	1		2	2		
	1	1	1	1	1		2		
		1	1		1		2		
	1	1		1	1		2		
	1		1	1					

- 점의 상,하,좌,우 빈 공간 중에 집의 내부 영역은 반드시 존재
- 연결된 빈 공간을 모두 검색하는 과정에서
  1. 게임 판 끝을 만나는 경우 : 외부
  2. 게임 판 끝을 만나지 않는 경우 : 내부

연속된 빈 공간 검색 방법 : dfs or bfs

감사합니다

