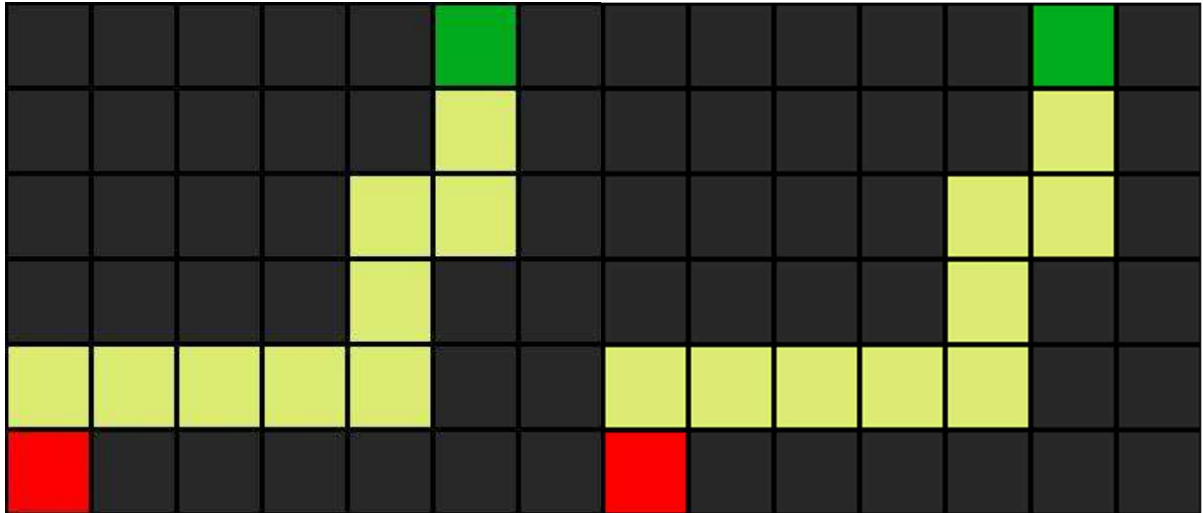
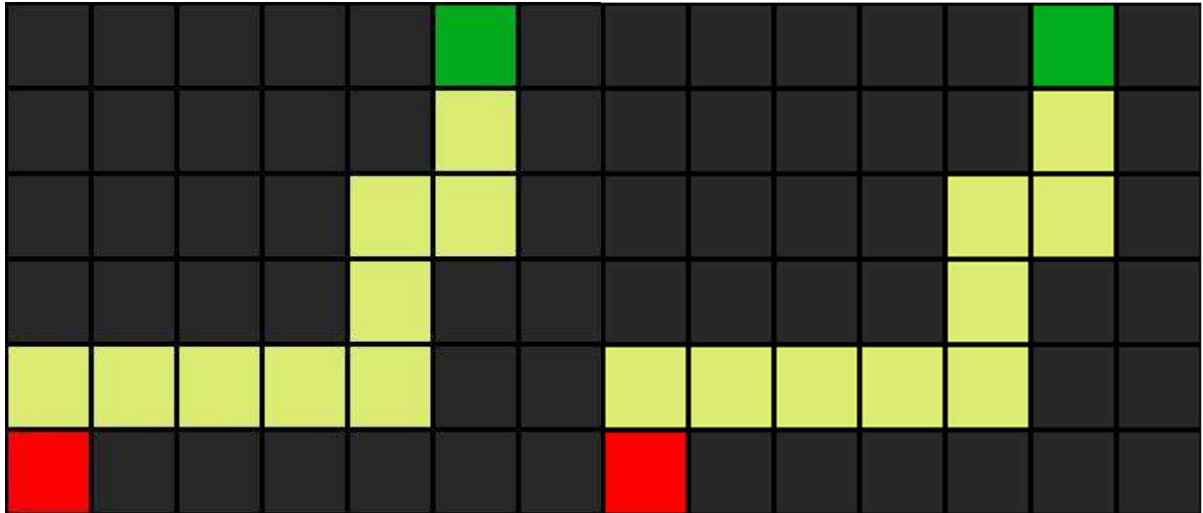


# Maze1

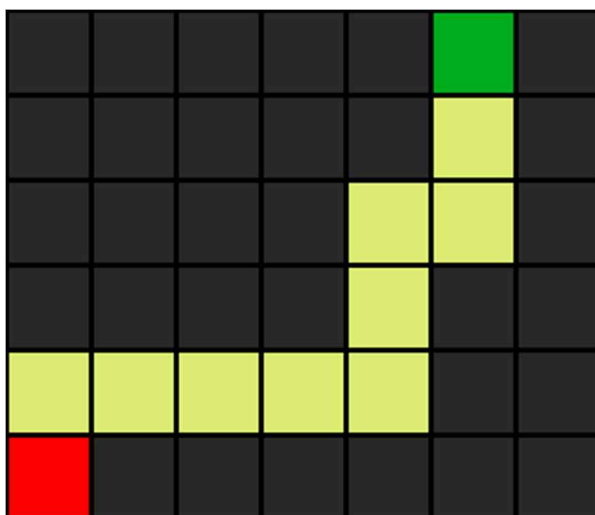
DFS(StackFrontier).png



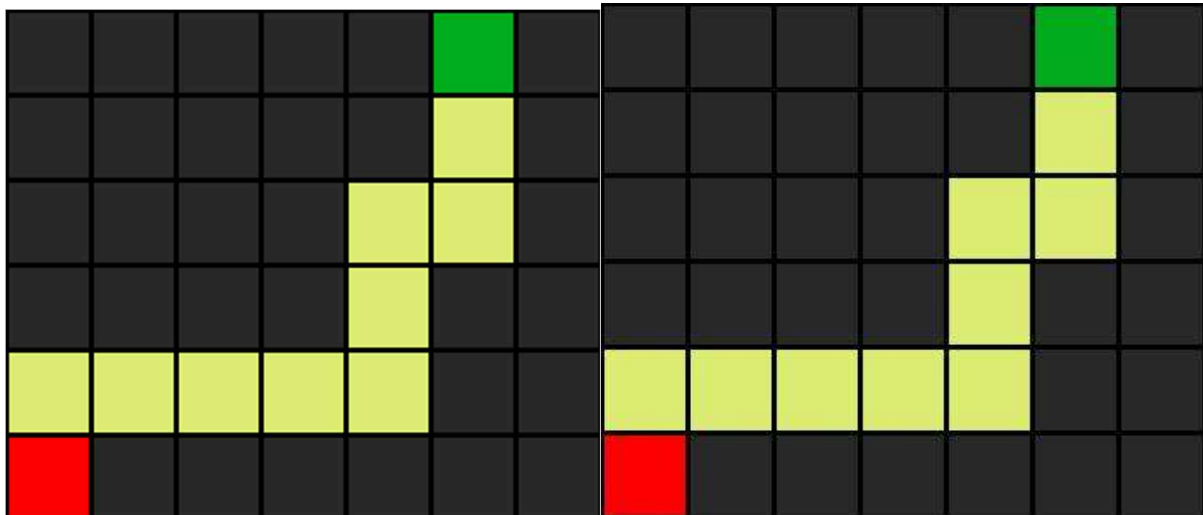
DFS(StackFrontier).jpg



BFS(QueueFrontier).png



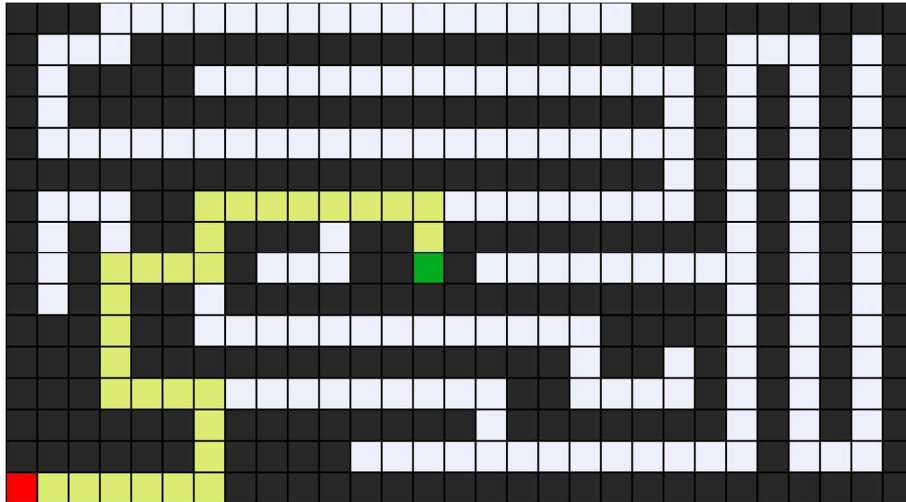
BFS(QueueFrontier).jpg



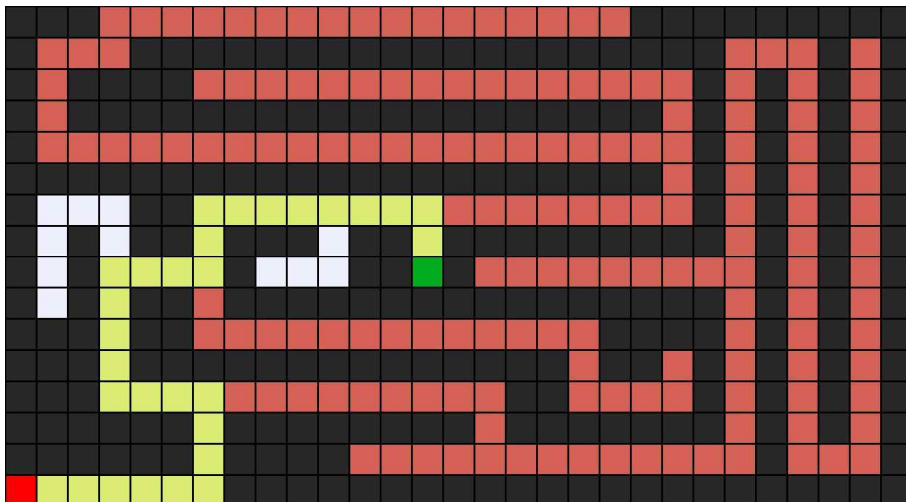
출발점에서 목표점까지 갈 수 있는 경로가 한가지 뿐이기 때문에 DFS, BFS 의 탐색 결과가 동일하고, jpg 에 방문한 노드도 표시되지 않았다.

## Maze2 - DFS

DFS(StackFrontier).png



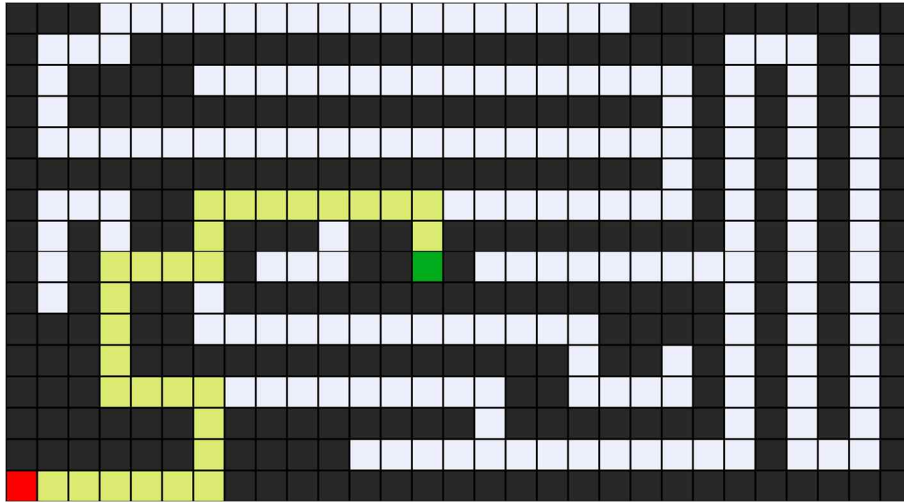
DFS(StackFrontier).jpg



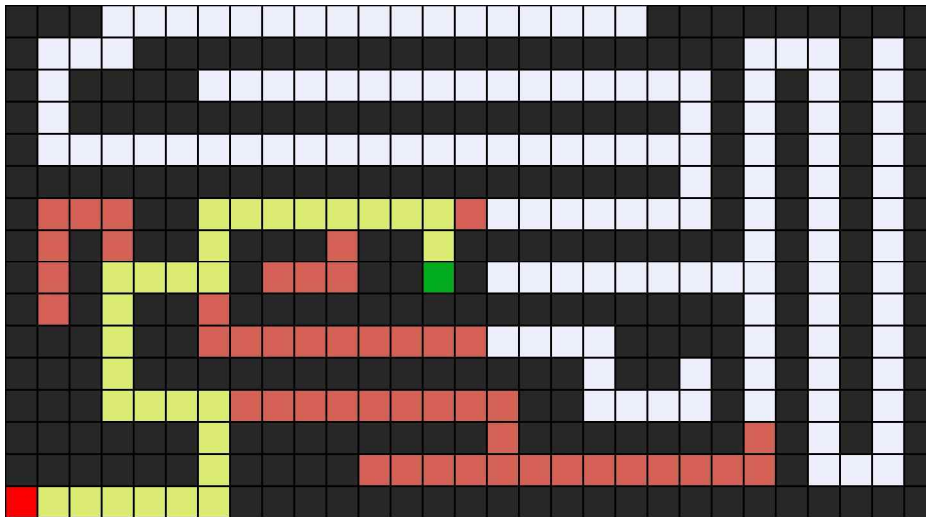
DFS png 와 jpg 비교 - 방문한 노드까지 표시한 jpg 에서는 붉은색으로 방문한 모든 노드가 표시되었다. 깊이우선탐색을 하였기 때문에 붉은색 노드들이 중간에 끊기지 않고 탐색 가능한 가장 깊은 곳의 노드까지 연결되어 있음을 확인할 수 있다.

## Maze2 - BFS

BFS(QueueFrontier).png



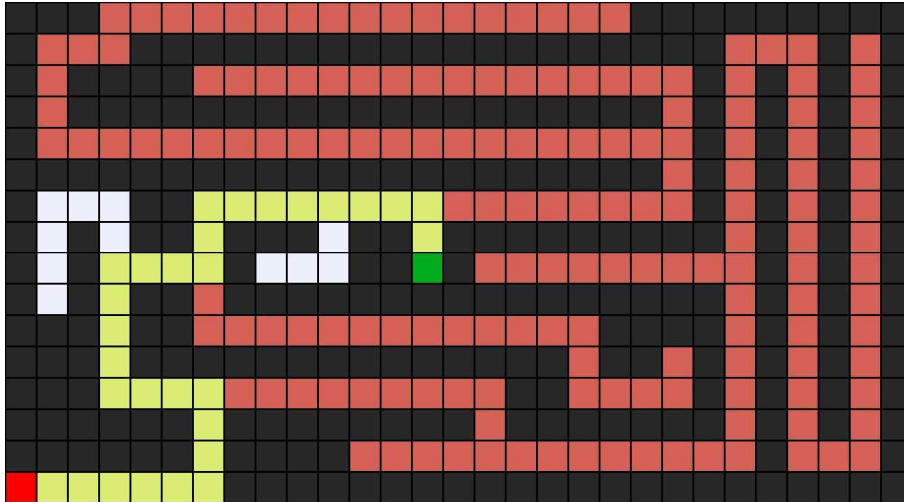
BFS(QueueFrontier).jpg



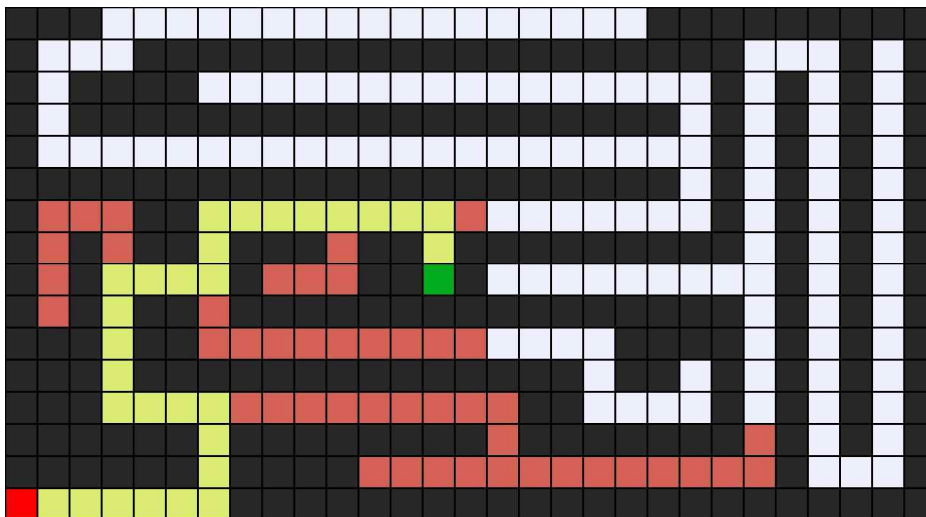
BFS png 와 jpg 비교 - 너비우선탐색을 하였기 때문에 붉은 색으로 표시된 방문한 노드들이 여러 곳으로 분포되어 있으나 막다른 길이 나올 정도로 깊은 곳을 방문하지는 않았음을 확인할 수 있다.

## Maze2 – DFS 와 BFS 비교

DFS



BFS

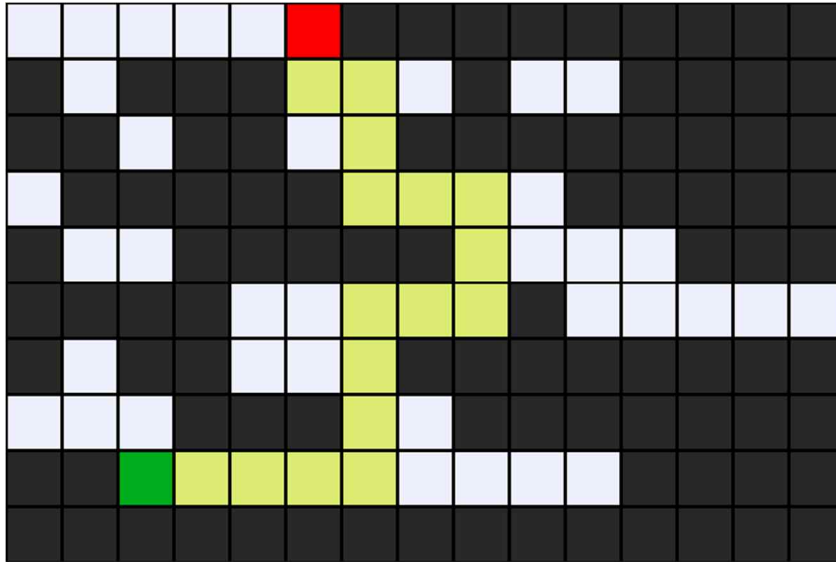


DFS 와 BFS 의 비교 – 두 탐색을 통해 찾은 경로는 같다. 하지만 Stack Frontier 를 사용한 DFS 는 깊이우선탐색으로 탐색할 곳을 정하고 나면 반드시 해당 경로가 막다른 길에 다다를 때까지 탐색하기 때문에 Maze2 에서는 QueueFrontier 를 사용한 BFS 보다 더 많은 노드를 방문했다.

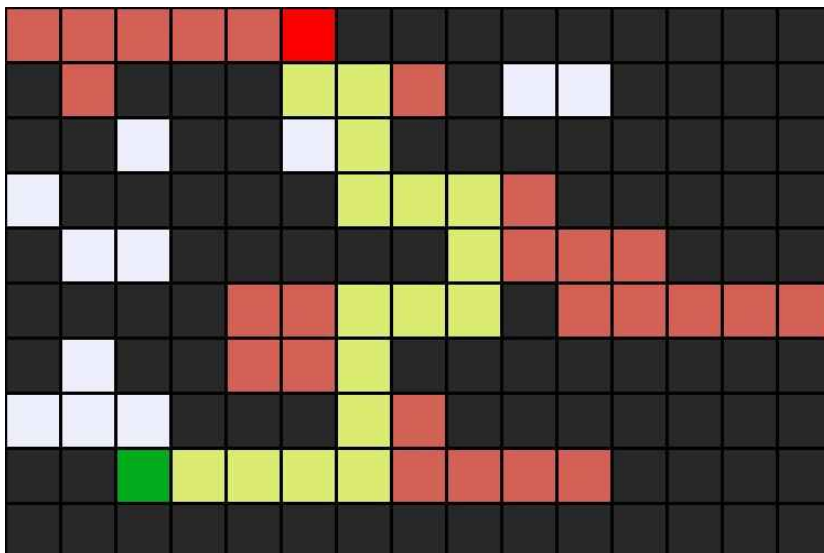
DFS(StackFrontier).jpg

# Maze4

DFS(StackFrontier).png

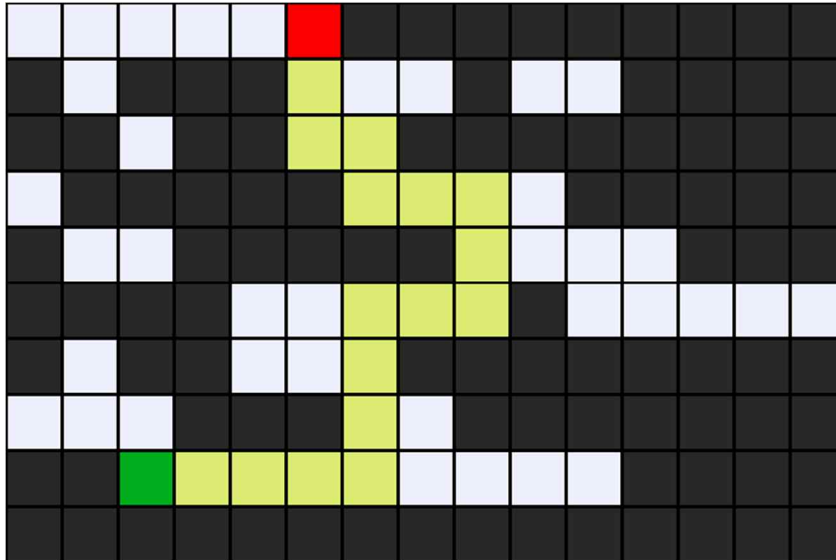


DFS(StackFrontier).jpg

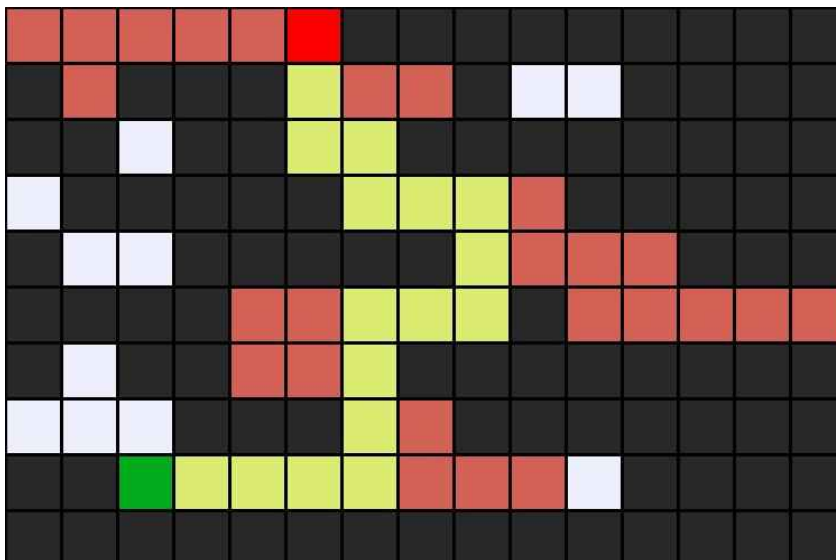


# Maze4

BFS(QueueFrontier).png



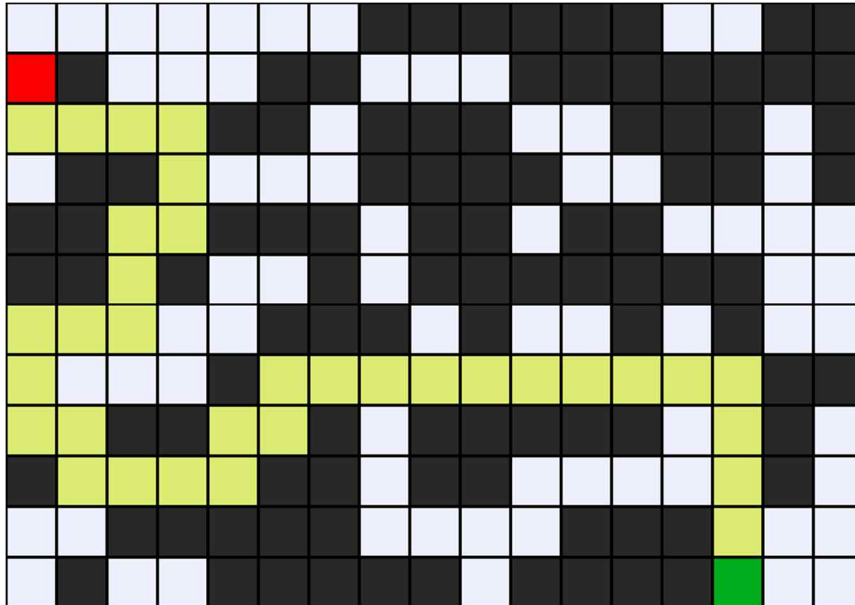
BFS(QueueFrontier).jpg



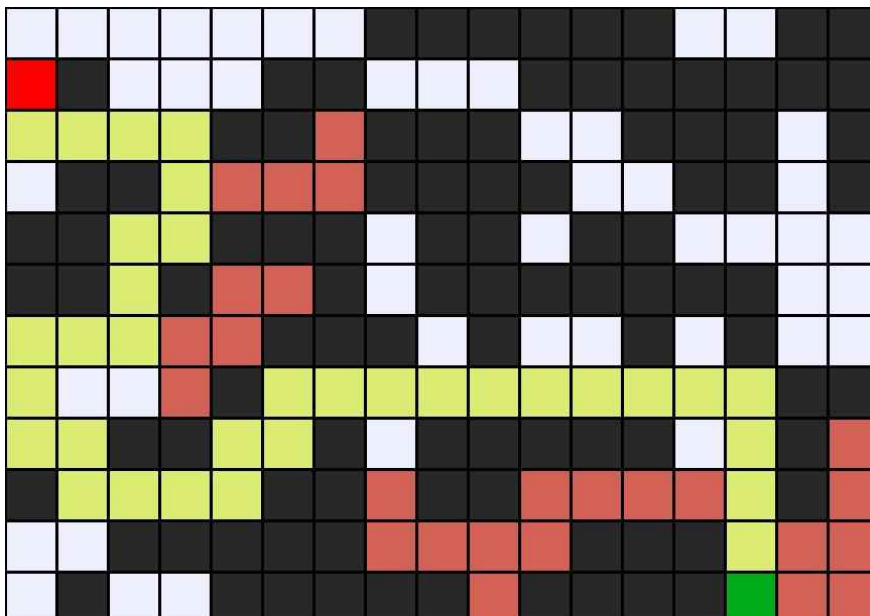
DFS 와 BFS 의 비교 – 방문한 노드는 대체로 비슷하나 경로의 일부가 다른 것을 확인할 수 있다.

# Maze5

DFS(StackFrontier).png



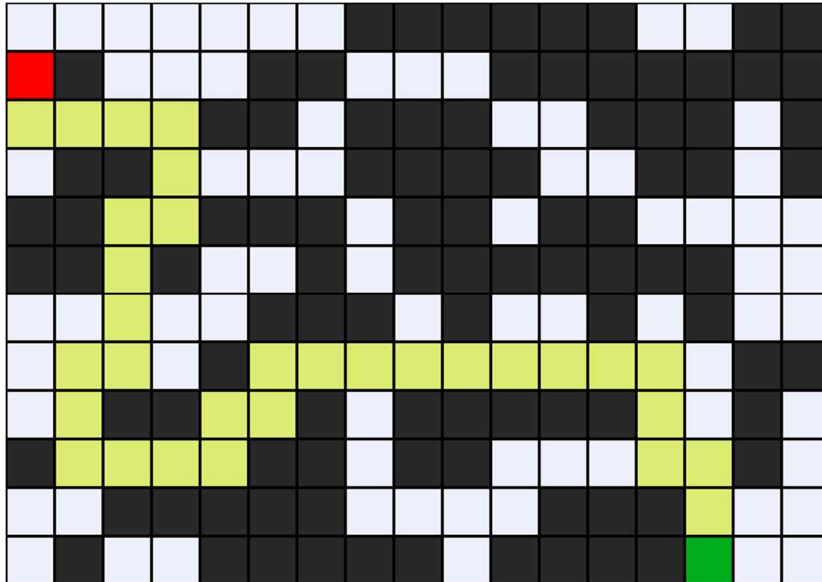
DFS(StackFrontier).jpg



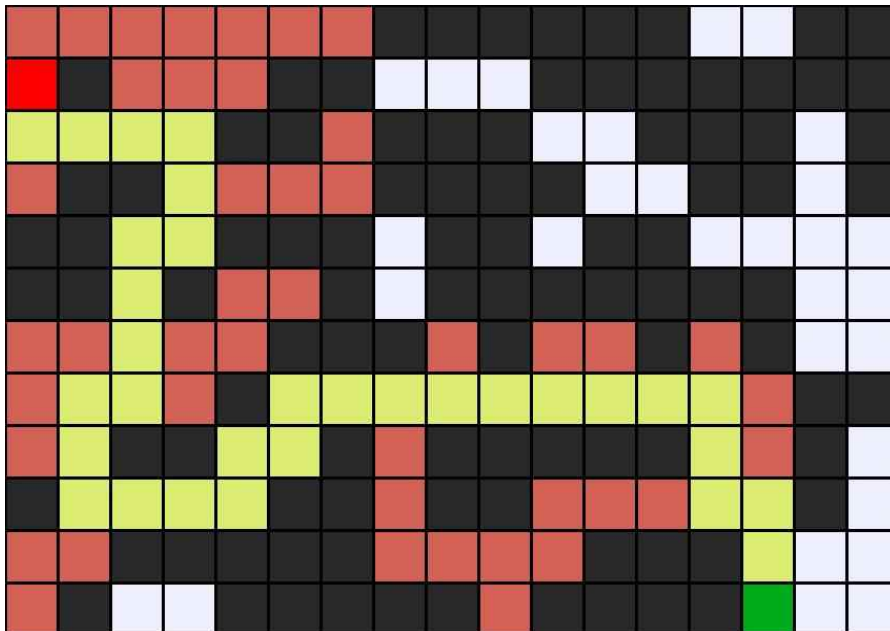


# Maze5

BFS(QueueFrontier).png



BFS(QueueFrontier).jpg



DFS 와 BFS 의 비교 - 두 탐색이 찾은 최종 경로에 차이가 있으며, 붉은색 경로(방문한 노드들)로 봤을 때 DFS 가 조금 더 효율적인 탐색을 한 것으로 보임.

## 최종 결과 비교

- ➔ BFS와 DFS의 결과 경로 차이가 있는 경우도 있고, 없는 경우도 있다. 또한 BFS가 더 적은 cost의 경로를 탐색하는 경우도 있고, DFS가 더 최적의 경로를 탐색하는 경우도 있다. 따라서 둘 중 어떤 탐색 방식이 더 효율적이며 optimal 경로를 찾는 성능면에서 나은지는 확정할 수가 없다