인공지능

과제 1 DFS, BFS, DFID

백은옥 교수님  
전민철, 조현진, 최승혁 조교님

2013011424  
성예닮  
컴퓨터공학부

1. 환경

운영체제: Windows 10  
언어: Java 8  
개발 IDE: IntelliJ IDEA

2. 알고리즘 요약

board[i] = j: i행, j열에 퀸이 놓여있다.

promising(board):  
  board의 어떤 두 여왕이 같은 열이나 대각선에서 만난다면:  
    False 반환  
  아니라면: True 반환

dfs(level, board, n):  
  level 이 n보다 같거나 크다면:  
    promising(board, n)이 참이면:  
      True를 반환;  
    아니라면: False를 반환;

for i, 0 to n:  
    board[level] = i;  
      dfs(level + 1, board, n)가 True라면:  
        True를 반환;  
  False를 반환;

bfs(level, board, n):  
  q.add({}) (q는 큐)

  level이 n보다 작을 때 반복:  
    현재 level의 모든 노드들에 대해:  
      p = q.pop  
      p.length가 n보다 같거나 크다면:  
        promising(board, n)이 참이라면:  
          True를 반환  
      s = p + e for e, 0 to n (s는 리스트)  
      s의 원소들 s[i]에 대해:  
        q.add(s[i])  
    level = level + 1

dfid(borad, n):  
  for level, 0 to n:  
    dfs(0, board, level)가 True라면:  
      True 반환  
  False 반환

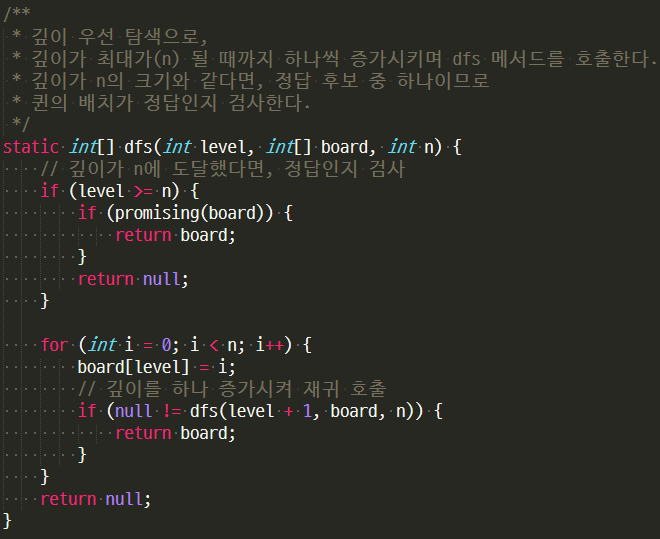
**3. 코드의 상세 설명**

promising



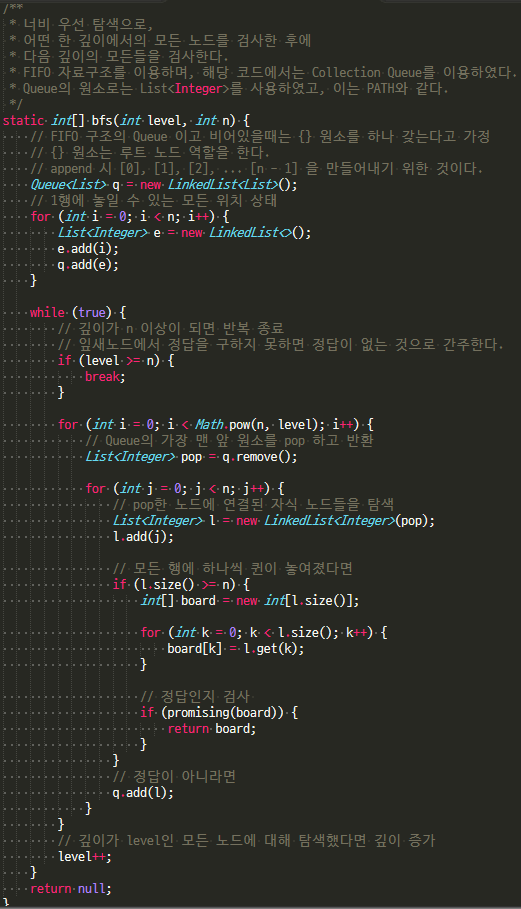
**현재 board에 놓인 각 여왕들이 같은 열 혹은 대각선에 존재하는지 검사하는 함수이다.**

**dfs**



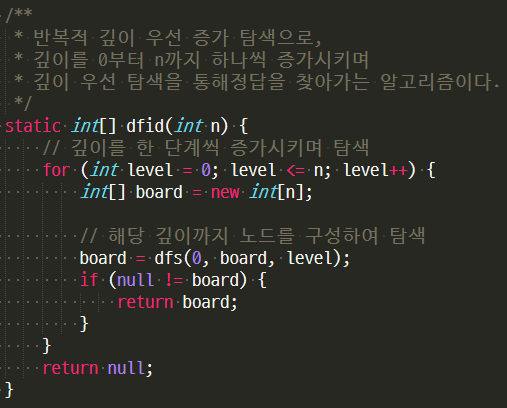
DFS를 수행하는 함수이다.  
DFS를 재귀적으로 호출하면서 깊이를 하나씩 증가시키는데,  
깊이의 수가 N과 같거나 크게 되면 가장 깊은 곳까지 탐색한 경우이고, 가장 깊은 곳에서 정답이 나오기 때문에, board가 정답인지 검사한다.  
재귀적으로 level을 N까지 증가시켰지만 정답이 없을 경우 거짓을 반환한다.

bfs



BFS를 수행하는 함수이다.  
큐 자료구조를 이용하여 각 level 별로 순회한다.  
예를 들어 큐에 [0], [1], [2], [3] 이 들어있다면, 큐의 FIFO 성질에 의해 [0]이 꺼내지고, [0]과 연결된 [0, 0], [0, 1], [0, 2], [0, 3]을 큐에 추가한다. 결과는 큐에 다음과 같은 노드들이 들어있다.  
[1], [2], [3], [0, 0], [0, 1], [0, 2], [0, 3]  
큐에 삽입되는 노드의 길이가 N과 같다면, 더 이상 큐에 추가하지 않고, 꺼내기만 한다.  
꺼내면서 해당 노드가 정답인지 검사한다.

dfid



Depth를 한 단계씩 증가시키며 탐색한다.  
Depth를 0부터 N까지 제한을 두며 DFS를 수행한다.  
마지막 Depth의 잎새 노드에서 정답이 나오므로, 결국 가장 마지막 Depth까지 수행해야 한다.

**4. 소스코드를 컴파일하는 방법**

**1) jar 파일 실행 방법**

**java -jar 2013011424.jar [퀸의 개수] [출력 파일의 절대 경로]  
예시) java -jar 2013011424.jar 8 D:AI**

**파일을 실행한 후에는 경로 아래에 output.txt 파일이 생성된다.**

**2) IntelliJ IDEA 프로젝트이기 때문에, assignment1 폴더를 다운로드하여 IntelliJ에서 Import 한다.**

**File -> New -> Project From Existing sources -> 다운로드 받은 경로 입력 후 OK**

**-> Create project from existing sources -> Project name 입력 후 모두 Overwrite**

**5. 기타 사항**

1) 출력 파일을 생성할 절대 경로에 스페이스가 포함되어 있다면 전체 경로를 “”(큰따옴표)로 감싸주어야 한다.  
예시) java -jar 2013011424.jar 4 "C:\Users\Yedarm Seong”

2) 출력 파일의 개행 문자가 CRLF가 아닌 LF기 때문에, Windows 메모장에서는 개행이 되지 않는다. UNIX 개행문자를 지원하는 에디터(Sublime text, atom 등)으로 출력 파일을 확인해야 한다.

3) DFS vs DFID  
실제 수행 시간은 DFID가 더 짧아야 하지만 처음으로 메모리에 변수가 올라가는 시간 등이 포함되어 처음 실행한 DFS가 더 오랜 시간이 걸리게 된다.