인공지능

과제 2 Local search

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 여왕이 놓여있다.

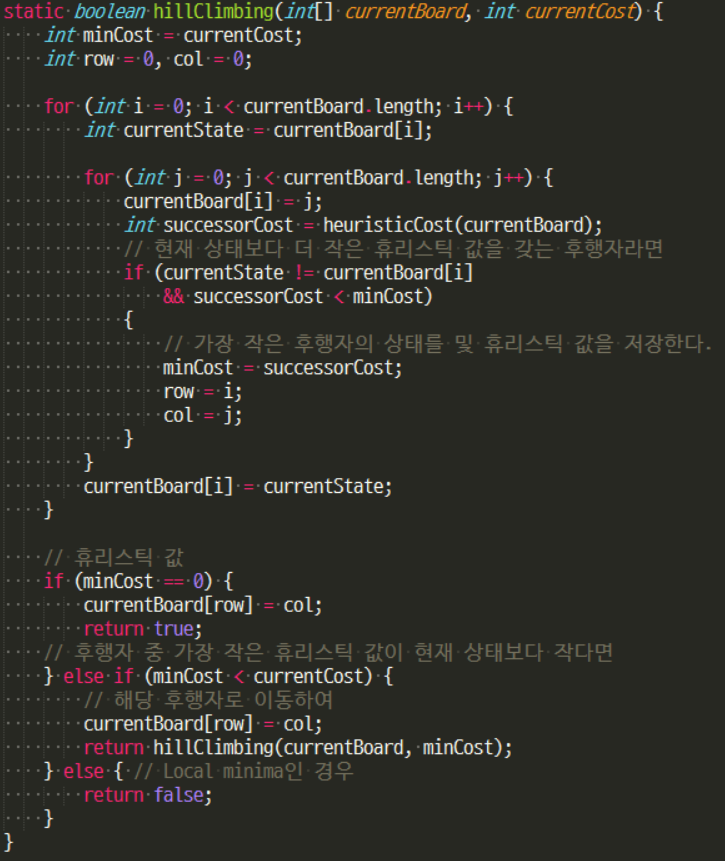
Heuristic function: 여왕이 가로, 세로, 대각선에서 만나는 횟수  
즉, 정답 상태는 Heuristic 값이 0인 상태이다.

Hill-climbing(현재 상태(초기 값은 random), 현재 상태의 휴리스틱 값):  
 현재 상태의 휴리스틱 값이 0 이라면 정답으로 간주하고 반환  
 반복  
 neighbor <- 모든 후행자 중 가장 휴리스틱 값이 작은 후행자  
 neighbor의 휴리스틱 값이 현재 상태의 휴리스틱 값보다 크다면  
 현재 상태를 반환(local minima)  
 현재 상태에 neighbor의 상태를 저장하고 Hill-climbing

Heuristic-cost(상태):  
 현재 상태에서 대각선이나 가로, 세로에서 공격할 수 있는 여왕의 쌍의 수를 반환

Random Restart: Local minima에 빠지면 restart를 한다.  
최대 N-queens의 N^2번 만큼 restart 한다.  
N^2번으로 5번의 실험 결과, N이 50이하인 경우 100% 찾아내었기 때문에 반복 횟수를 N^2로 정했다.

**3. 코드의 상세 설명**

**Hill climbing**

현재 상태에서의 휴리스틱 값과 모든 후행자들의 휴리스틱 값을 비교하여,  
후행자들 중 휴리스틱 값이 가장 작은 값이, 현재 상태에서의 휴리스틱 값보다 작으면  
해당 후행자로 이동하여 다시 Hill climbing을 반복한다.  
local minima인 경우 False를 반환하고 Restart를, 정답인 경우 True를 반환한다.

Heuristic cost  


인자로 받은 board의 휴리스틱을 계산한다.  
하나의 행은 하나의 여왕만 놓을 수 있으므로, 행간의 비교와 대각선 비교만 필요로 한다.

**4. 소스코드를 컴파일하는 방법**

**1) java 파일을 다운로드 받아 실행한다.  
컴퓨터의 환경이 encoding=UTF-8, line separator=UNIX(LF)인 경우 소스 코드를 올바르게 컴파일 하고 출력 결과를 올바르게 확인할 수 있다.**

**2) IntelliJ IDEA 프로젝트이기 때문에, assignmen2 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) 모든 상태가 Global minima이자 Local minima인 경우, 혹은 N^2번의 반복 restart에서도 Local minima에서 빠져나오지 못하고 정답을 찾지 못한 경우에는 출력 파일에 “Couldn't escape the local minima”을 출력하고 Location에는 가장 마지막 Local minima 상태를 출력한다.