

# Artificial Intelligence

## Programming Assignment 2

Local Search: Hill Climbing

due : 10/5 토요일 자정 (24:00)

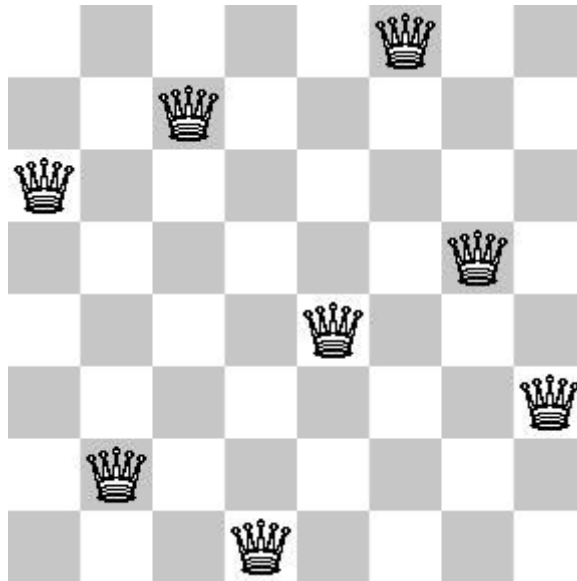
### 1. 개요

수업시간에 배운 Hill Climbing을 활용하여 'N-Queens'문제를 해결한다.

### 2. 과제 설명

#### a. 'N-Queens'문제란?

- i. 'N x N'의 체스판이 있다.
- ii. N개의 Queen이 서로 공격하지 않아야한다.  
(Queen은 가로, 세로, 대각선에 있는 말들을 모두 공격할 수 있다.)
- iii. N=8일 때의 답 예시



#### b. 'N-Queens' 문제 해결

'N-Queens' 문제를 'Hill climbing' 방법을 사용하여 풀어본다. 이 때, 각각의 Search 에서 첫 번째로 찾아진 답을 출력한다. 답을 출력할 때에는, 0~(N-1) 까지의 Column에 있는 각 Queen의 row 위치를 출력한다. (ex. '2-a-iii'의 그림의 답은 2,6,1,7,4,0,3,5 이다.)

#### c. 구현 제약사항

- 1) 각 Column에 있는 Queens의 초기 위치는 '랜덤'하게 지정한다.
- 2) objective function을 각자 정의 한 후 구현 한다. **Local optimum에 갇힐 경우 restart** 하거나 **random walk**를 통해서 빠져 나올 것.

d. 입력 형식

모든 입력은 Argument로 받는다. 첫 번째 Argument는 N을 입력받고, 두 번째 Argument는 결과 출력 파일의 '절대 경로'를 입력 받는다. 예시는 아래와 같다.

```
java -jar 2017103064.jar 4 D:\Assignment2
```

e. 출력 형식

결과 출력 파일 이름은 "resultN.txt"로 한다. 예를 들어 N=8일 때, 파일 이름은 "result8.txt"이다. 모든 공백은 Space bar 한칸으로 한다.

예) result8.txt

```
>Hill Climbing  
3 5 0 4 1 7 2 6  
Total Elapsed Time: 0.002
```

f. 보고서 필수 기재 내용

- i. 알고리즘 및 objective function에 대한 설명
- ii. 실행 화면 출력
- iii. 결과에 대한 분석

ex> 평균적으로 random walk or restart를 몇 번?

3. 사용 언어

- a. Java
- b. C

단, C는 테스트 환경을 리눅스 gcc 컴파일러로 하고, 개별적인 환경에 대해서는 구체적으로 작성해야 함 (보고서에 추가). 실행이 안될 경우 0점 처리 될 수 있음 (**되도록 Java로 구현하기 바람**).

4. 코드 및 보고서 제출 방법

- a. "assignment2" 디렉토리 생성 후 해당 디렉토리 안에 소스코드 위치시킨 후 commit & push 할 것
- b. 보고서는 GitLab Wiki에 "assignment2" 페이지 생성 후 작성
- c. GitLab clone 또는 push시에 인증 오류가 난다면, GitLab web page 에서 파일 첨부를 통해 제출 할 것.
- d. GitLab을 통한 자세한 제출 방법은 포털 내 인공지능 강의실 홈페이지 공지 참조

5. 주의 사항

- a. 코드가 완성되면 반드시 GitLab에 push 해야 함
- b. 실행 파일이 아닌, 소스코드를 제출 할 것
- c. 코드에 주석 작성 필수 (주석 없으면 0점 처리)

- d. 컴파일 오류시 0점 처리
- e. 제출 기한 초과시
  - i. 기본 20% 감점 후 매 24시간마다 20%씩 추가 감점  
(ex. 50시간 초과시 60% 감점)
  - ii. 72시간 이후 : 0점 처리