

Artificial Intelligence

Programming Assignment 1

DFS, BFS, DFID

due : 9/21 토요일 자정 (24:00)

1. 개요

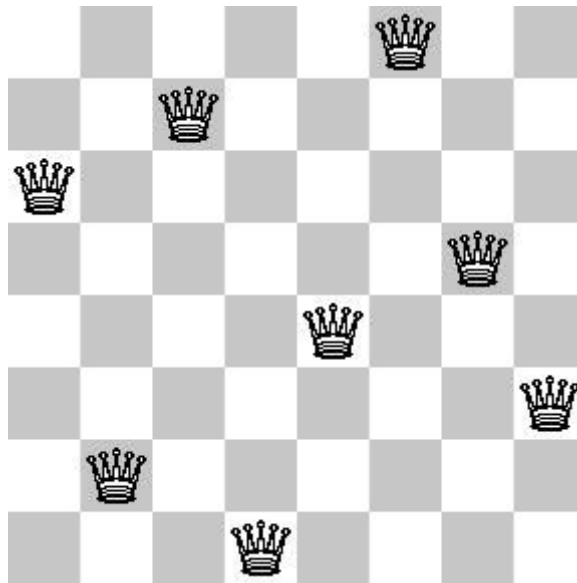
[Iterative deepening depth-first search](#)

수업시간에 배운 DFS, BFS, DFID를 토대로 'N-Queens' 문제를 해결한다.

2. 과제 설명

a. 'N-Queens' 문제란?

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



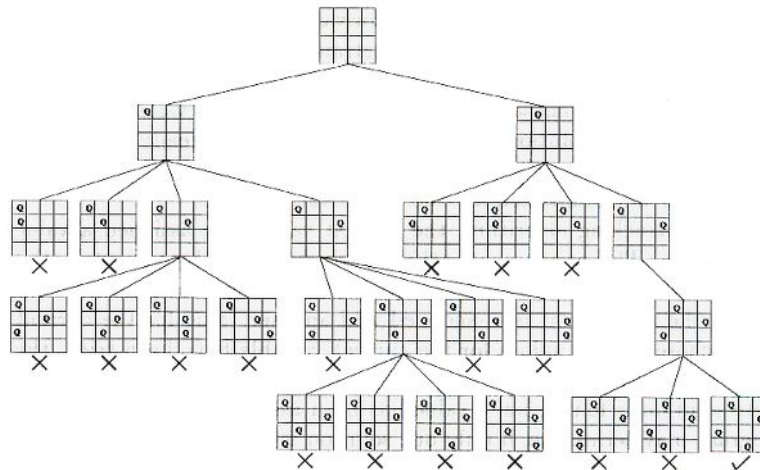
b. 'N-Queens' 문제 해결

'N-Queens' 문제를 'DFS, BFS, DFID Search' 방법을 각각 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.)

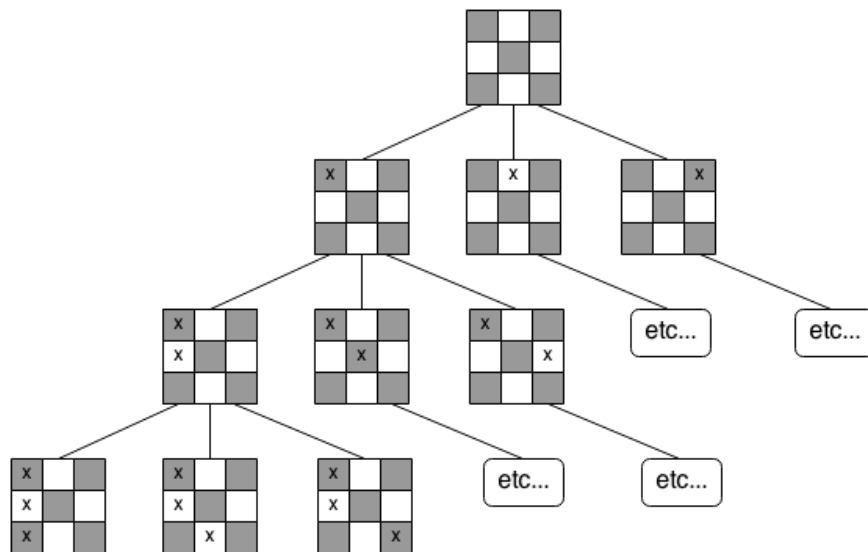
가지치기X 우선 다 넣고 마지막에 개수를 다 채우면 검사 시작할 것!

c. 주의 할 점

Search를 할때 <그림1>과 같이 Pruning하지 않고, <그림2>과 같이 모두 expanding 시켜준다.



<그림1>



<그림2>

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.

```
D:\W2017Assistant\WAI\Assignment1>java -jar 2017103064.jar 4 D:\W2017Assistant\WAI\Assignment1
```

e. 출력 형식

결과 출력 파일 이름은 “resultN.txt”로 한다. 예를 들어 N=5일 때, 파일 이름은 “result5.txt”이다. 모든 공백은 Space bar 한칸으로 한다.

예) result7.txt

```
>DFS
Location : 6 4 2 0 5 3 1
Time : 0.04300000000000003

>BFS
Location : 0 2 4 6 1 3 5
Time : 0.324

>DFID
Location : 6 4 2 0 5 3 1
Time : 0.232
```

답이 없다면, “Location : ~ “ 대신에 “No solution” 으로 출력하여 준다.

예) result3.txt

```
>DFS
No solution
Time : 0.0

>BFS
No solution
Time : 0.0

>DFID
No solution
Time : 0.0
```

3. 사용 언어

- a. Java
- b. C

단, C는 테스트한 환경은 리눅스 gcc 컴파일로 하시고, 개별적인 환경에 대해서는 구체적으로 작성해야 함 (보고서에 추가). 실행이 안될 경우 0점 처리 될 수 있음 (**되도록 Java로 구현하길 바람**).

4. 코드 및 보고서 제출 방법

- a. “assignment1” 디렉토리 생성 후 해당 디렉토리 안에 소스코드를 위치시킨 후 commit & push 할 것.
- b. 보고서는 GitLab Wiki에 “assignment1” 페이지 생성 후 작성
- c. 자세한 제출 방법은 포털 내 인공지능 강의실 홈페이지 공지 참조

5. 주의 사항

- a. 코드가 완성되면 반드시 GitLab에 push 해야 함
- b. 코드에 주석 작성 필수 (주석 없으면 0점 처리)
- c. 컴파일 오류시 0점 처리
- d. 제출 기한 초과시

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