Ahsanullah University of Science and Technology
Department of Computer Science and Engineering (CSE)

CSE4108: Artificial Intelligence Lab, Spring 2018

Lab Group: B2 Offline: 4 Topic: Local Search

ATTENTION

Read the documentation/specification provided thoroughly before you have started doing your assignment. Everything we want have been stated here properly. :)

1 Introduction

You will write two local search algorithms to solve a Queen-Rook placement problem in a chess-board.

2 Problem Statement

You will have a 8×8 chessboard. You will be given 5 pieces of Queens and 3 pieces of Rooks.You have to place them in such a way that those pieces do not attack each other. Two things for sure: neither you can put any two of them in same column nor in same row.

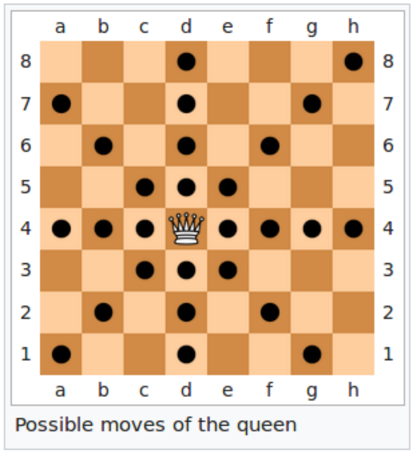
3 Moves

3.1 Queen

The queen can be moved any number of unoccupied squares in a straight line vertically, horizontally, or diagonally, thus combining the moves of the rook and bishop (see Figure 1). The queen captures by occupying the square on which an enemy piece sits.

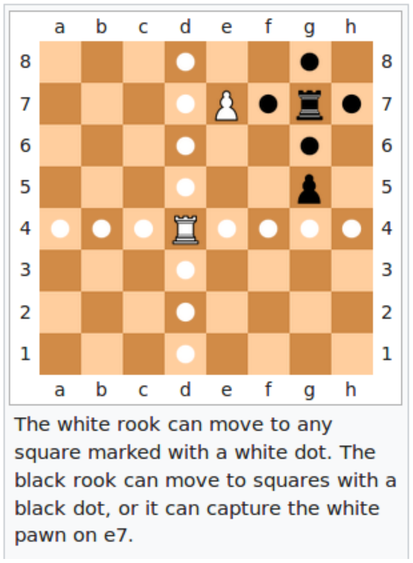
3.2 Rook

The rook moves horizontally or vertically, through any number of unoccupied squares (see). As with captures by other pieces, the rook captures by occupying the square on which the enemy piece sits.



Possible moves of the queen

Figure 1: Queen's moves.



The white rook can move to any square marked with a white dot. The black rook can move to squares with a black dot, or it can capture the white pawn on e7.

Figure 2: Rook's moves.

3.3 Solution strings

As per the codes shown in reading materials, you can think of a solution string as string consisting of 1-8, 5 q's and 3 r's. So it will be a 16 character string. From a permutation of this string, extract the digits 1-8 and characters (r and q's) separately. Then the first character and the first digit will denote the first piece in first row, and so on. Say , a string is "56qrq124qq7r8rq3". So, the digits extracted in order are "56124783" and characters extracted in order are "qrqqqrrq". it means, the placement of pieces in different rows are - (row 1, column 5 , queen), (row 2, column 6 , rook), (row 3, column 1 , queen), (row 4, column 2 , queen), (row 5, column 4 , queen), (row 5, column 7 , rook), (row 6, column 8 , rook), (row 8, column 4 , queen).

3.4 fitness

How good a combination string is.

How to apply it in this problem

You can find the number of attacking pairs as the "unfit"-ness function. There are $8C2 = 28$ possible pairs. The ideal placement has no attacking pair in it. So you can calculate fitness function as (28−number of attacking pairs). You can design your own fitness function too. But remember, the ideal placement will always have the highest fitness and the more ideal a placement is near to, the more fitness it will have. For example,if there are two attacking pairs, then fitness will be $28 - 2 = 26$.

4 Output

Print the whole board using *,q,r and the fitness of the board.

5 Algorithms

Here, by roll number, we denote the last three digits of the ID. $Roll\_Number\%3 == 0$ have to implement **Hill Climbing and Simulated Annealing**. $Roll\_Number\%3 == 1$ have to implement **Steepest ascent Hill Climbing and Simulated Annealing**.
$Roll\_Number\%3 == 2$ have to implement **Hill Climbing and Steepest ascent Hill Climbing**.

6 Language

You can use any language. But try to do it in Python if you can- it will be a good practice.