REVIEW EXERCISE 03

Question 1. The *N*-queens problem requires you to place *N* queens on an N × N chessboard such that no queen attacks another queen. (A queen attacks any piece in the same row or column or diagonal). Here are some important facts:

- The states are any configurations where **all** *N* queens are on the board, one per column.
- The ***moveset*** includes all possible states generated by moving a single queen to another square in the same column. The function to obtain these states is called the ***successor*** function.
- The heuristic function ***h(state)*** is the number of **attacking** pairs of queens.

a) Consider *N*=4. How many states are there in total? Explain your answer.

4^4 = 256 states. There are 4 queens and each queen can be in any of the four squares in its column.

b) For each state, how many successor states are there in the moveset? Explain your answer

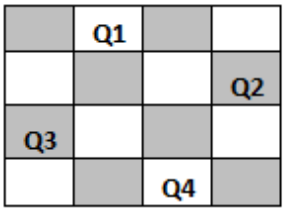
3 x 4 = 12 states. Each queen can move to any of the rest three squares in its column. Only one queen is moved in a successor

c) What value will the heuristic function ***h*(*state*)** return for state *S* shown aside? Explain your answer.

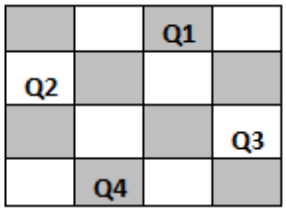
5

d) Use some hill-climbing variant that can lead to a solution. Draw the search tree from S (Only draw the branches that lead to a solution; for each node on the tree, write down its ***h*( )** value).

There are multiple ways that lead to solutions. Following are the only two configurations which exist for the 4 queen's problem.

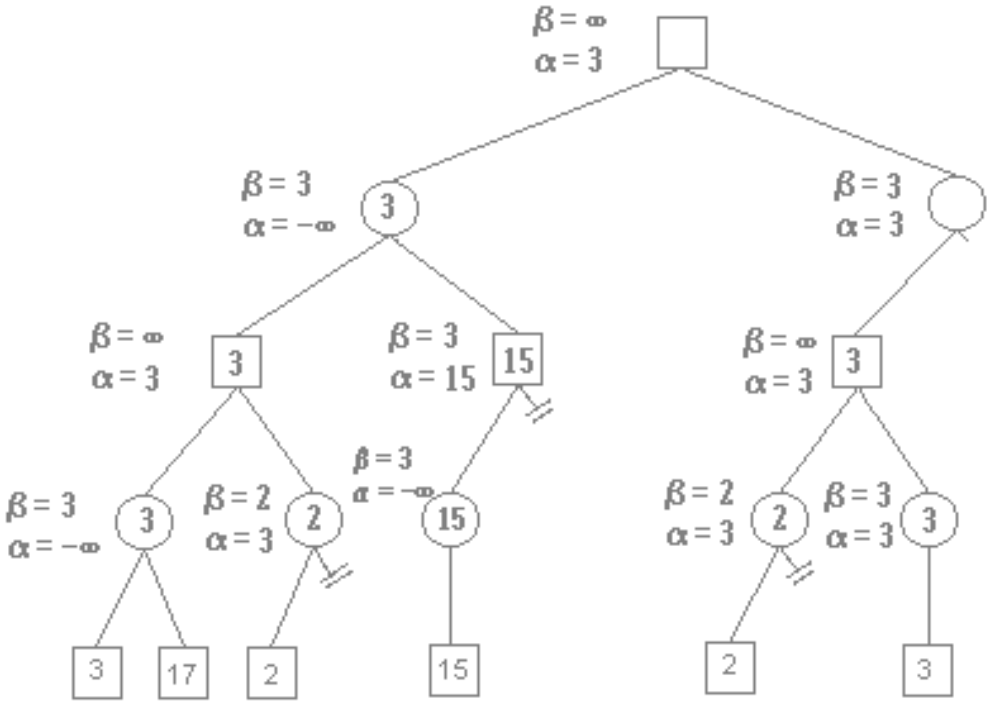


Solution 1



Solution 2

Question 2. Consider the game search tree shown below, in which a square denotes MAX's move and a circle for MIN's move . Write down the utility value for each node of the tree using Minimax algorithm. Show where alpha-beta pruning occurs by crossing out corresponding branches. Briefly explain for each case.



Refer to the full explanation in http://web.cs.ucla.edu/~rosen/161/notes/alphabeta.html