

Implementing A* Search Algorithm

- 1 Initialize by placing the queens randomly, with one queen per column, along with initializing priority queue & heuristics.
- 2 Dequeue the node with the lowest f value from priority queue; if node has no conflicts, return possible solution.
- 3 Generate Succession move by moving the queen to any row within its column.
Calculate f value ($g + h$) for every step by calculating no. of conflicts.
- 4 Push the successors into priority queue and add g to h in priority queue for each succession where g = cost to reach the current state from start state.
- 5 Continue expanding & exploring Successors until solution is found.

Algorithm for 8 queen using Hill climbing

1 Initialization

Start with a random configuration where a queen is placed in each column

2 Iteratively Improve.

- Compute the heuristic for the current state
- If heuristic value is zero, the solution is found

3 Generate neighbouring states:

For each queen generate the neighbouring states by moving it to every possible row within its column & calculate the heuristic for each new state

4 Choose the best neighbour

Select the neighbour with local heuristic

If no neighbour state improves the current state, terminate

5 Repeat until solution is achieved