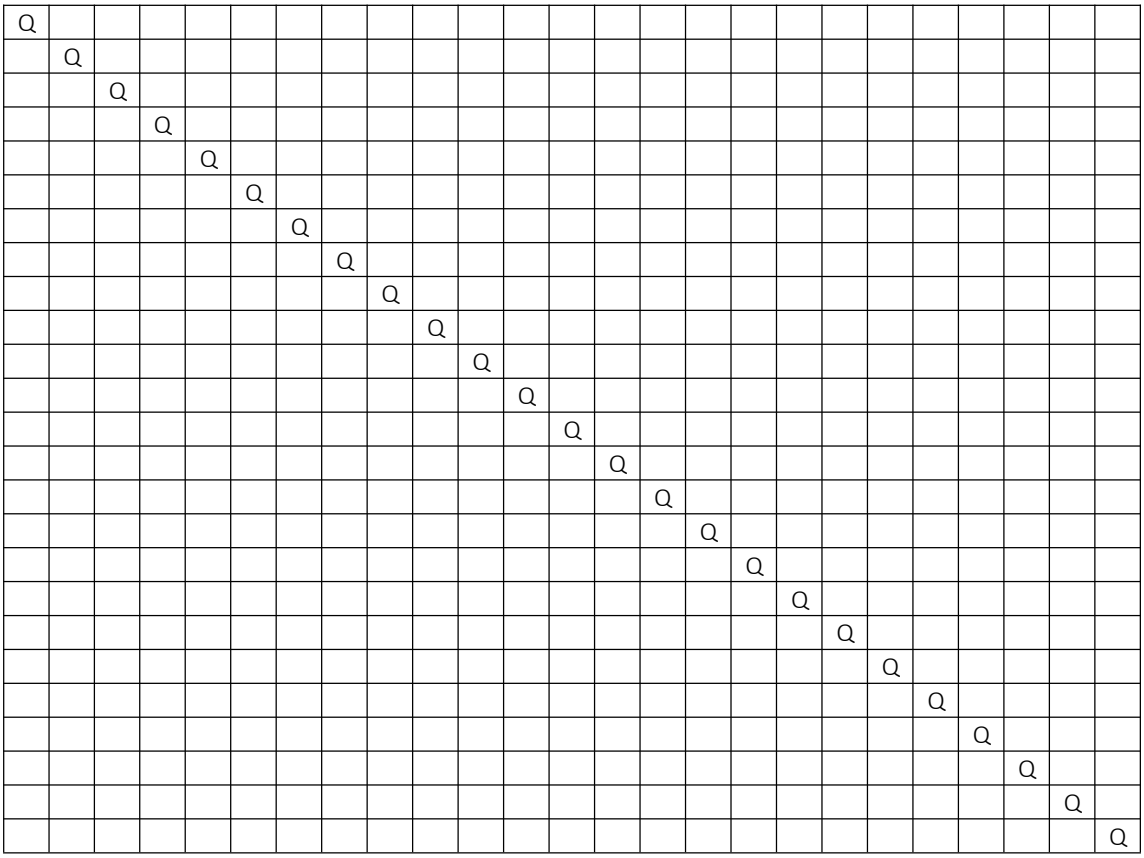
1 (Programming) Simulated Annealing

Considering our initial state as the one shown in the figure below.



I have used Simulated Annealing method to solve the above problem.

- Score Function : I have used the score function f(s) as no of attacking queens . i.e f(s) = Number of Attacking Queens .For Example in the initial state the score of queen present in column 25 is f(s) = 24 . I am considering the minimum value of f(s) to move in the neighbourhood.

- Neighbourhood : I have taken the neighbourhood of the queen as rows of its corresponding columns . For Example for a queen belonging to column 25 of the chess board the neighbourhood of the queen is the rows present in column 25 .

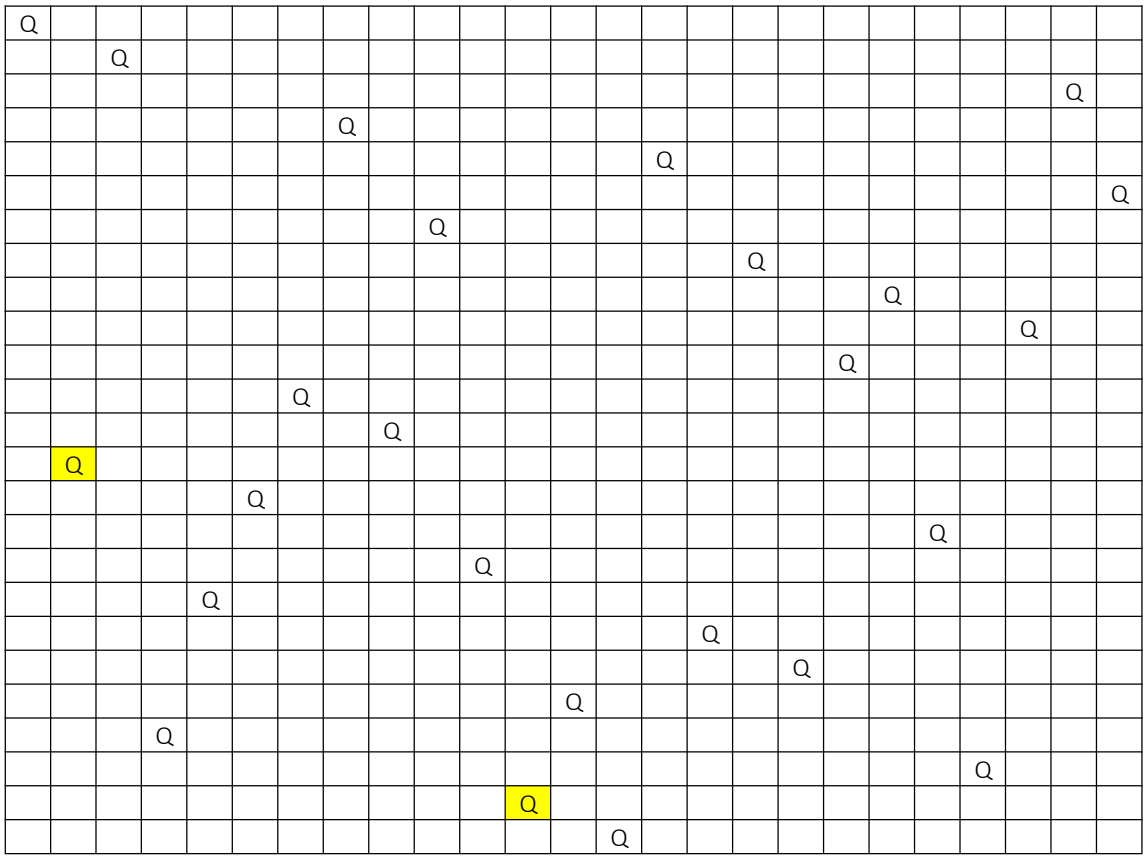
- Approach : I have moved from right to left and for each queen I have checked its neighborhood and If there is a point in its neighbourhood for which score function is better I have moved my queen to that point , else if there is no such better state in the neighbourhood I have accepted a state with a probability p . Probability p is defined as

$$p = \exp\left(\frac{-(f(s)-f(t))^2}{\text{Temp}}\right)$$

Where I had done the process for three values of temperature Temp = 10 , 5 ,1 . For each case I have accepted a state with a probability p and I have decreased the temperature as $T_1 = T_0 * 0.9$ and I have stopped the process after reaching a threshold probability of value 0.1 .

Results :

I have obtained the best state as follows .



I have obtained the best state using the approach stated above .
In my best state the no of conflicting queens = 2 , the conflicting queens are marked in yellow.

- I have done the process for three values of temperature when the temperature is high the number of states explored is also high . For my particular case after a certain point we can see that f(s) = f(t) for every point in the neighbourhood of the queen therefore the difference f(s) - f(t) = 0 and the probability is 1 , so now the algorithm has changed to First Choice Hill Climbing . For a very low value of temperature the no of explored states is also low and the point where f(s) = f(t) is reached very quick , since we did not explore many states the no of conflicts is higher for low value of temperature . For my case I have obtained the best state when I have taken a higher temperature , For higher temperature the no conflicts in the best state is 2 and for lower the no of conflicts in best state is 4 . So its better to have a higher temperature as no of states will be explored will be more and the probability of finding a best state is also high when compared to low temperature.

- So far for my case Simulated Annealing has the best performance when the temperature is high . Vanilla hill climbing algorithm got stuck in a local optima because of bad initial start and greediness of the algorithm . I have used

Stochastic Hill Climbing Algorithm next , Stochastic Hill Climbing is better compared to Vanilla Hill Climbing since it consider the 2nd and 3rd best states as well but Stochastic Hill Climbing is still greedy because it doesn't consider the worst state the chances of solving the problem is better for Stochastic compared to vanilla if the initial point is good . For Simulated Annealing since we are accepting the worst state with a probability our chances of getting struck in a local optima is very less and if the temperature is high we will explore more and more states and thus we will be tending to the best state . Comparing all the three algorithms I feel Simulated Annealing has a better chance of solving the problem compared to the other two since it overcome the disadvantages faced by Vanilla Hill Climbing and Stochastic Hill Climbing and I got the better result using Simulated Annealing.