Group 7 technical document

<u>Group members and contributions:</u>

Yichang Lu, student number: 20096224, contributions: writing “maping” function

Jiajun Ren, student number: 20109684, contributions: making technical document and testing the results of the algorithm.

Zijian lu, student number: 20090405, contributions: writing “mini_conf” and “countingfailure” functions

Xiaoyue Ai, student number: 20119662, contributions: writing “mini_conf” function.

Xiaohan Xie, student number: , 20090392, contributions: debugging and editing “solve” function .

Conclusion of the algorithm

This algorithm read the input file and get the values of n which are the numbers of queens and side lengths of the chessboard. Then it put n queens on the chessboard and records the locations of each row, column, and diagonal of every queen. According to the result of conflict calculation by the locations of queens, it adjusts every queen’s position for returning the result with the minimum number of conflicts. To be noticed, this algorithm only returns one of the solutions instead of all solutions.

Here are more details of “solve(board_size)” method and other supporting parts:

1. **Method: solve(board_size):**
   The “solve” method gets n as its parameter. Then “solve” calls mini_conf(grid_size,10000) to return a single list of integers that represent the solution of the N-queens problem. It is called run.py to show the answer to the N-queens problem.

2. **Function: maping(n):**
   The function maping forms a random matrix with n Queens on each row and column. “loc[x] = y” means there is a Q on row x colunm y, and “rldiag “means the number of queens on right to left diagonal, and “lrdiag” shows the number of queens on left to right diagonal. “maping” returns a list that represent queens’ positions the function returns the location of the Queen and the number of queens in each diagonal.

3. **Function: mini_conf(n,maxstep):**
   The mini_conf function is used to obtain the minimum conflict of the row in which the column needs to be changed by using the mini-conflict method and to move the queen to the position

of the minimum conflict. Input is n is the size of the Nqueens board, and maxstep is the number of times we can repeat the search for conflict. First, It runs the maping function to get the initial state of a chessboard with modify position. Then count the position and the times of conflict using the countingfailure function. Then choose the row with the highest conflict to change to the position of the queen. To change the position, we add up the Conflicts number on the diagonal of the column, as a result, we can figure out the number of conflicts in each column of the Queen on that row, and therefore the smallest one. After the Queen is moved to an appropriate position, the conflict-affected by the front and rear positions of the Queen is reset. We can allow a maximum number of searches, and when the number is large enough that no results have been found, we can give up.

4. Function: countingfailure (list):

This function gets the result of the lists from maping(n) function. According to the lists, it will check the conflicts through the row, column, left to right diagonal, and right to left diagonal. At the same time, it counts the times of fail. Then it returns two lists - the position of the number of failed rows and the number of failed rows.