

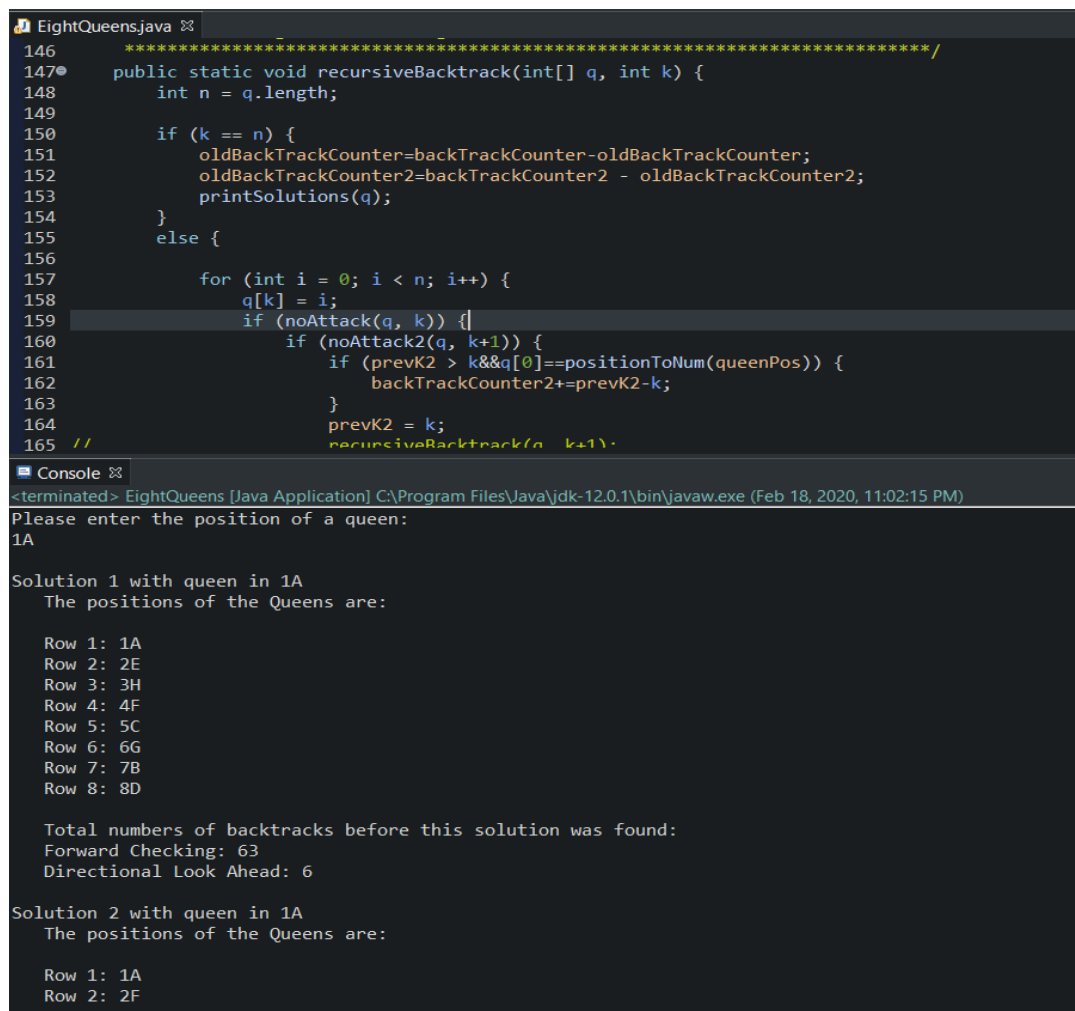
Jessmer John Palanca

README: 8-QUEENS PROBLEM

This program gives all the solutions for the 8-queens problem. It will display the placement (row x column) of the 8-queens and the number of backtracks of taken to find the solution. To run this program:

1. Right-click the EightQueens.java.
2. Select Run As > Java Application.
3. A prompt will display in the console asking for the placement of the first Queen on the first row.
4. Enter the placement of the first Queen.
5. The solution for the first Queen's problem will display to the console.

Below is the sample output when running the program, given 1A as the placement of the first Queen:



```
EightQueens.java
146 *****
147 public static void recursiveBacktrack(int[] q, int k) {
148     int n = q.length;
149
150     if (k == n) {
151         oldBackTrackCounter=backTrackCounter-oldBackTrackCounter;
152         oldBackTrackCounter2=backTrackCounter2 - oldBackTrackCounter2;
153         printSolutions(q);
154     }
155     else {
156
157         for (int i = 0; i < n; i++) {
158             q[k] = i;
159             if (noAttack(q, k)) {
160                 if (noAttack2(q, k+1)) {
161                     if (prevK2 > k&&q[0]==positionToNum(queenPos)) {
162                         backTrackCounter2+=prevK2-k;
163                     }
164                     prevK2 = k;
165                     recursiveBacktrack(q, k+1);
166                 }
167             }
168         }
169     }
170 }
171 *****

Console
<terminated> EightQueens [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Feb 18, 2020, 11:02:15 PM)
Please enter the position of a queen:
1A

Solution 1 with queen in 1A
The positions of the Queens are:

Row 1: 1A
Row 2: 2E
Row 3: 3H
Row 4: 4F
Row 5: 5C
Row 6: 6G
Row 7: 7B
Row 8: 8D

Total numbers of backtracks before this solution was found:
Forward Checking: 63
Directional Look Ahead: 6

Solution 2 with queen in 1A
The positions of the Queens are:

Row 1: 1A
Row 2: 2F
```

The program has 2 functions that checks for the problem constraints, noAttacks (used for Directional Arc Lookahead) and noAttacks2 (Forward Checking). The positionToNum function converts the user input to an int (Queen placement by column number). The printSolutions function prints the solutions to the Queens' placements and the backtracking counter for both forward checking and the directional arc lookahead. The recursiveBacktrack function is the main recursive function that implements backtracking algorithm. Inside this function, we implemented the forward checking and directional arc lookahead. The counter for backtracks is also implemented inside the recursive function.