Mathematical Approach for N-Queens Problem with Isomorphism

Sanjay Jain

Department of Mathematical Sciences, SPC Government College, Ajmer
Affiliated to M. D. S. University Ajmer, Ajmer-305001, India

E-mail drjainsanjay@gmail.com

Chander Shakher

Research Scholar, Mahatma Gandhi University, Meghalaya

E-mail csghorela200921@gmail.com

Abstract

The N-Queens problem is commonly used to teach the programming technique of backtrack search. The N-Queens problem may also be used to illustrate the important concept of isomorphism. Here, we study how the N-Queens problem can be used as a vehicle to teach the concepts of isomorphism, transformation groups or generators, and equivalence classes. We also indicate how these ideas can be used in a programming exercise and compute non-isomorphic solutions.

Keyword: N-Queens, Isomorphism, Transformation, Non-isomorphic solutions.

1. Introduction

The 8-Queens problem or more general N-Queens problem is often used to explicate backtracking in computing courses. In this paper we want to point out that N-Queens can also be used as a vehicle for teaching the ideas of isomorphism and transformation groups. First we provide the brief introduction of terminology used in this work.

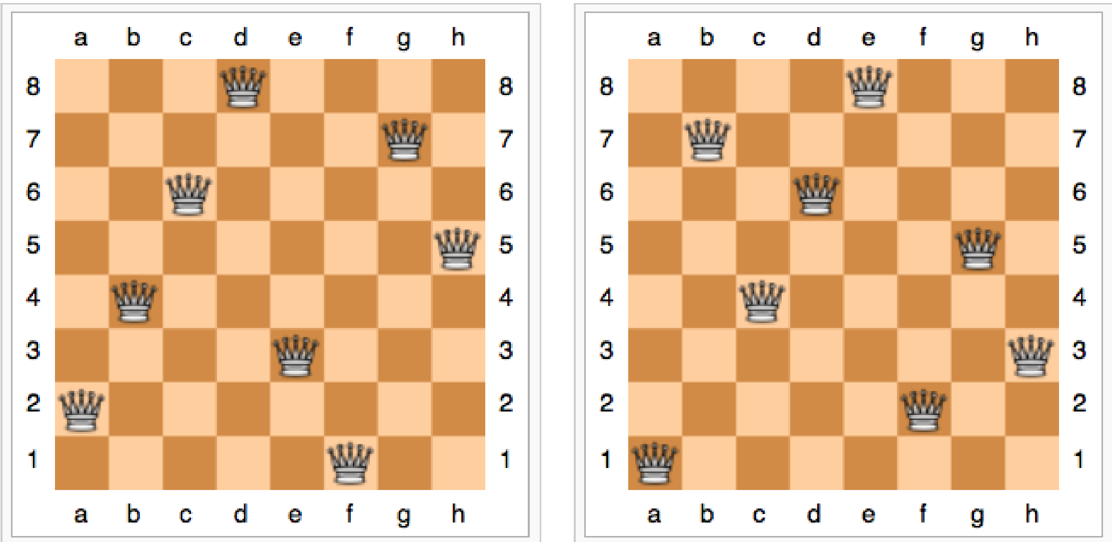
- **Backtracking**

Bitner & Reingold (1975) studied backtrack programming problem. “Backtracking is a general algorithm for finding all (or some) solutions to some computational problem that incrementally builds candidates to the solutions, and abandons each partial candidate c ("backtracks") as soon as it determines that c cannot possibly be completed to a valid solution.”

- **Eight queens puzzle**

The eight queens puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens attack each other. Thus, a solution requires that no two queens share the same row, column, or diagonal. The eight queens puzzle is an example of the more general ***n*-queens problem** of placing *n* queens on an *n* × *n* chessboard, where solutions exist for all natural numbers *n* with the exception of *n* = 2 and *n* = 3. This puzzle provides a classical example in the field of computation for studying recursion, depth-first search and backtracking.

Clay (1986) and Goldsby (1987) provides solutions in different manner for 8 queens problem. The eight queens puzzle has 92 **distinct** solutions. If solutions that differ only by symmetry operations (rotations and reflections) of the board are counted as one, the puzzle has 12 **unique** (or **fundamental**) solutions.



Unique solution 1 Unique solution 2

Figure 1(1.1 & 1.2): Two solutions to the 8-Queens problem.

- **The N-Queens Problem**

The objective in the N-Queens problem is to place N queens on an N X N chessboard such that no two queens can attack one another under the normal rules of chess. A queen may attack any other piece lying along the row, column, or diagonal containing the queen. Bruen & Diyon (1975) also studied N-Queens problem. Demirors & Tanik (1991) works on queens with magic squares.

The N-Queens problem is often stated as: Can N queens are placed on an N X N chessboard so that no queen can attack another queen. In this form the answer is easy: if N not belongs to

{2,3} say Yes. Further, it is easy to find a non-attacking placement for the queens. Abramson & Yung (1989) and Hoffman, Loessi & Moore (1969) also provides solutions in different manner for N-Queens problem.

Two solutions to the 8-Queens problem are shown in Figure 1. Since any solution to N-Queens must have exactly one queen per row (and column) of the chessboard, a solution such as Figure 1.1 can be expressed as the vector (2,4,6,8,3,1,5,7) i.e. the row which the queen in each column occupies, Similarly, solution 1.2 can be notated as (1,7,4,6,8,2,5,3).

The construction of an enumerating backtracking program seems to be necessary to find all the non-attacking placements. Even finding the number of such placements seems to be difficult.

The point here is that the N-Queens problem gives us a chance to discuss isomorphic and non-isomorphic solutions. Further, by a small addition to the standard N-Queens program, we can produce a program which outputs only the non-isomorphic solutions. In addition, this exercise will also introduce the use of transformation groups.

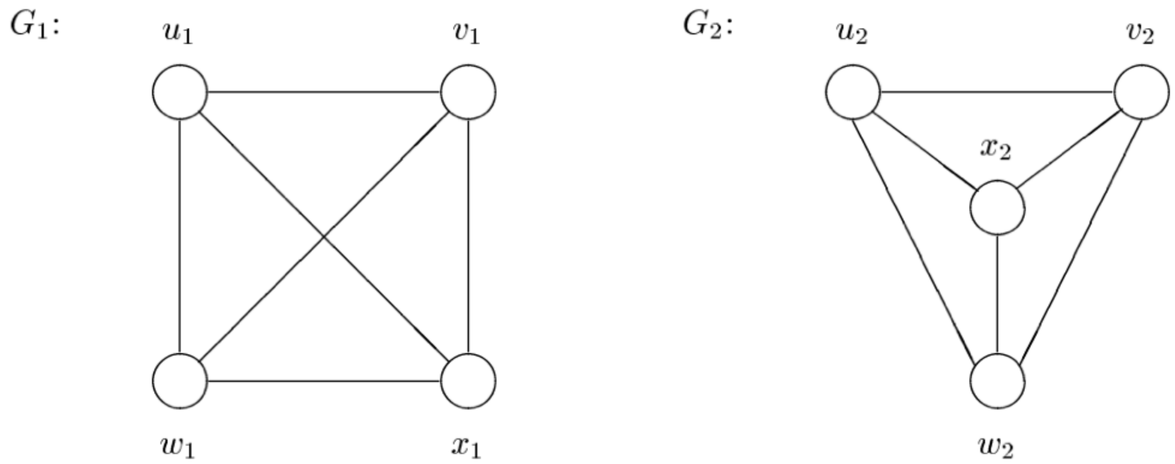


Figure 2: Two Isomorphic Graphs.

- **Isomorphism**

Two objects are isomorphic when they have equal forms, that is, in some sense the objects are identical. The sense is, of course, important. For example, a metal paper clip and a plastic paper clip are isomorphic when I want to clip papers together, but they are not isomorphic when I want to play with my magnetic paper clip holder. In more complicated cases, the objects are made up of parts and the sense of sameness includes some of the interrelationships between the parts. For example, I might consider bees and mosquitoes as isomorphic in producing red welts on humans, but when I consider the body parts, the bee and the mosquito are not isomorphic because their business ends are different.

In mathematical study, the idea of isomorphism is usually introduced in algebra, For example, the field of real numbers is isomorphic to the field of complex numbers with zero imaginary part, or two semi groups (S, +) and (G,*) are isomorphic when there is an invertible function

$$h: S \longrightarrow G \text{ so that } h(s_1+s_2) = h(s_1)*(s_2) \text{ for all } s_1 \text{ and } s_2.$$

Computer scientists are usually introduced to isomorphism in the context of graphs. Two graphs G1 and G2 are isomorphic if there is an invertible function h which maps each vertex of G1 to a vertex of G2 so that adjacent vertices of G1 map to adjacent vertices of G2 and vice versa, as in Figure 2.

- **Isomorphism and the N-Queens Problem**

For solutions to the N-Queens, we want the renaming to preserve the relationships between the queens. What transformations should be allowed? Clearly mirror image, mentioned above, should be included, but it seems that several mirror images are possible. The obvious ones place a mirror parallel to one side of the board, but what about reflecting across one of the diagonals of the board? Other allowed transformations should include rotations by multiples of 90º (π/2 radians). Obviously one could combine rotations and mirror images to get other transformations. This combining is function composition since each transformation takes a solution as input and gives a solution as output. Function composition is associative. Associativity means that if T1, T2, and T3 are three transformations and T1.T2 is the composite transformation obtained by applying T2 and then T1, then the following equation holds (T1.T2).T3 = T1.(T2.T3).

Further, the composition of any allowed transformations always gives an allowed transformation, so the set of transformations is closed under composition. There is a special transformation, 'do nothing to the board', which is the identity transformation. Finally, each transformation can be undone, that is, each transformation has an inverse. Each mirror image is its own inverse. For any rotation by a multiple of 90 degree, applying the rotation 3 times will give the inverse of the rotation. So for any sequence of rotations and mirror images, there is an inverse which can be formed by taking the inverses of the transformations in reverse order.

A set with an operation that is associative, closed, and has an identity and an inverse is, of course, a group. In fact, the allowed transformation for the N-Queens is the group of transformations which transform a square into itself. Consider a board with corners labelled A, B, C, and D in clockwise order. When a transformation is applied, the corner labelled A can be mapped to any of 4 positions, and the next corner clockwise after A must be either B or D. So there are 4*2 = 8 different transformations. Chandra (1974) studied for independent transformations.

The set of transformations for N-Queens is a dihedral group. A dihedral group Δ_n is defined as the group of symmetries of a regular polygon P_n of n sides. Elements of Δ_n can be obtained by the operations of rotation R through 360º /n, and the operation reflection M about some side:

$I \Longleftrightarrow R \Longleftrightarrow R^2 \Longleftrightarrow R^3 \ldots R^n$

Using the operations of reflection (M)

$M \Longleftrightarrow RM \Longleftrightarrow R^2 M \Longleftrightarrow R^3 M \ldots R^n M$

For N-Queens the appropriate group is $\Delta 4$. Using the operations of reflection (M), and 90º rotation (R), one solution can be transformed into seven other solutions.

- **Generating Non-isomorphic N-Queens Solutions**

An interesting problem is finding the number of non-isomorphic solutions for each value of N, as well as a method of quickly generating these non-isomorphic solutions. It might seem easy to find the number of non-isomorphic solutions to N-Queens by finding the total number of solutions and dividing by 8. Unfortunately this doesn't work because there are solutions which can be mapped to themselves by some of the transformations.

For example, the solution (2,4,6,8,3,1,5,7) to 8-Queens (Figure 1.1) is transformed to itself after two rotations:

2,4,6,8,3,1,5,7 $\Longleftrightarrow$ 3,8,4,7,1,6,2,5 $\Longleftrightarrow$ 4,2,8,6,1,3,5,7 $\Longleftrightarrow$ 4,7,3,8,2,5,1,6

Using the operations of reflection (M)

7,5,1,3,8,6,4,2 $\Longleftrightarrow$ 5,2,6,1,7,4,8,3 $\Longleftrightarrow$ 7,5,3,1,6,8,2,4 $\Longleftrightarrow$ 6,1,5,2,8,3,7,4

Below is a table showing the total number of solutions to N-Queens, as well as the number of non-isomorphic solutions to the N-Queens problem for various values of N:

N	Total number of solution	Number of non isomorphic solution
0	0	0
1	1	1
2	0	0
3	0	0
4	2	1
5	10	2
6	4	1
7	40	6
8	92	12
9	352	46
10	724	92
11	2680	341
12	14200	1787
13	73712	9233
14	365596	45752
15	2279184	285053

2. Mathematical Approach

While we want to be able to count the number of non-isomorphic solutions, we also want to find one solution for each isomorphism class. Two solutions are in the same isomorphism class exactly when some allowed transformation transforms one solution to the other. It is probably worth mentioning at this point that being isomorphic is an equivalence relation, and reminding students that an equivalence relation is reflexive, symmetric, and transitive. As an easy exercise you can ask the students to find which group property implies reflexive, which group property implies symmetric, and which group property implies transitive.

Since there are only 8 transformations to consider, a simple way to find non-isomorphic solutions is to maintain a set of the non-isomorphic solutions found so far, generate all solutions, and then determine if applying any of the 8 transformations gives a solution already in the set. If each of the 8 transformations gives a solution not in the set, then the new solution should be added to the set of non-isomorphic solutions.

Unfortunately this method has the drawback that there may be a very large (more than exponential) number of non-isomorphic solutions in the set. So comparing a new solution with the non-isomorphic solutions can take a very long time. Luckily, there is a shortcut. In the backtracking algorithm, the first queen is placed in the first allowed square in the first column, then the second queen is placed in the first allowed position in the second column, and so forth. This means that the solutions will be generated in order if we consider each solution of N-Queens as a base N+1 number. Hence a solution is isomorphic to a previously found solution if and only if one of the 8 transformations produces a solution which is less than the present solution.

3. Mathematical Formulation of equation

The less than relation may be stated as:

Let $S_1 = (a_1,a_2,a_3\ldots,a_n)$ and $S_2 = (b_1,b_2,b_3\ldots,b_n)$

which are two solutions to the N-Queens, so that each $a_i,b_i \in \{1,2,\ldots,n\}$.

Then $S_1 < S_2$ iff $(a_1a_2a_3\ldots a_n) < (b_1b_2b_3\ldots b_n)$,

where: $(a_1a_2a_3\ldots a_n) < (b_1b_2b_3\ldots b_n)$ iff $a_1<b_1$, in the usual ordering $1< 2 <\ldots< n$

or, $a_1 = b_1$ and $(a_2a_3\ldots a_n) < (b_2b_3\ldots b_n)$

…
…
to bottom out the recursion, we have $(a_n) < (b_n)$ iff $a_n < b_n$ in the usual ordering.

So given a solution S, one should add S to the set of non-isomorphic solutions when for each of the seven non-identity transformations $T_1,T_2,\ldots,T_7$,

$S \leq T_i(S)$, for i= 1,2,…,7.

4. Programming Algorithm

Biernat (1993), Clapp, Trevor & Volz (1986), Cockayne & Hedetniemi (1986), Erbas & Tanik (1992), Sosic & Gu (1990) and many researchers gives different algorithms for different queens problem. Here we prepared an algorithm to check isomorphic and find generator from a queens problem and non-isomorphic solutions.

```
/*   check_if_isomorphic: If the solution is an isomorphic to a previously generated solution,
return TRUE. Return FALSE if solution is new.

*/

BOOLEAN check_if_isomorph(int original[MAX])

{

int i;    int transformed[MAX];

BOOLEAN iso_flag;

for (i = 0; i < size; i++)          //make a copy of the solution vector

transformed[i] = original[i];

for (i=0;i<7;i++)         //generated the 7 transformation

{

if(i!=3)

Rotate(transformed);
    else
Mirror_image(transformed);    iso_flag = compare_vector(original, transformed);

if (! iso_flag) return TRUE;

}    return FALSE;

}
```

The generators R and M are not the only possible generators. Let F be the transformation which flips the square across its counter-diagonal. Then F and M are a set of generators for Δ4, and FMFMFMF is a sequence of these generators which will generate the whole transformation group. Showing that the previous statement is true would b e a reasonable exercise to see if your students have followed the development. You also might want them to decide which set of generators is easier to program.

In several texts, the fact that the queen in the first column never has to be placed in the second half of the column if one is only interested in non-isomorphic solutions is mentioned, as Horowitz and Sahni (1979).

> "Observe that for finding in equivalent solutions the algorithm need only set
> X (I) = 2,3,…,[n/2]."

Unfortunately this has been widely misinterpreted by users to mean that this restriction alone is sufficient to generate only non-isomorphic solutions. We hope that the above description has been sufficient to explain the actual scenario.

5. Conclusion

The major point of this note is that N-Queens is a good example which is typical of combinatorial enumeration problems. The typical features are:

1 solutions are generated by backtracking algorithms

2 ordering on solutions is imposed by the generating algorithm

3 desire for non-isomorphic solutions

4 group of transformations indicating which solutions are isomorphic

5 group expressible by simple generators

6 "on-the-fly" non-isomorphism test by applying a sequence of generators to a solution and checking that all the transformed solutions are ≥ the solution in question.

Cull & De Curtins (1978) studied Knight's Tour Revisited, which is another problem with similar characteristics of N queens problem. Knight's Tour Problem (where the transformation group consists of rotation, reflection, and "take a path backwards") is open for further research in isomorphism approach.

References

[1] Abramson, Bruce and Yung, Moti, (1989) "Divide and Conquer under Global Constraints: A Solution to the N-Queens Problem", Journal of Parallel and Distributed Computing, 6, 649-662.

[2] Bell, Jordan, and Brett Stevens. (2009) "A Survey of Known Results and Research Areas For N-Queens." Discrete Mathematics 309.1: 1-31.

[3] Bessière, Christian, Et Al. (2005) "Asynchronous Backtracking Without Adding Links: A New Member In The Abt Family." Artificial Intelligence 161.1: 7-24.

[4] Biernat, Martin J., (1993) "Teaching Tools for Data Structures and Algorithms", ACM SIGCSE Bulletin, 25, 9-12.

[5] Bitner, James R., and Reingold, Edward M., (1975) "Backtrack Programming Techniques", Communications of the ACM, 18, 651-656.

[6] Bruen, A., and Dixon, R., (1975) "The n-Queens Problem" Discrete Mathematics, ACM SIGCSE Bulletin, 12:393-395.

[7] Chandra, A. K., (1974) "Independent Permutations as Related to a Problem of Moser and a Theorem of Polya." Journal of Combinatorial Theory, Series A, 16, 111–120.

[8] Clapp, Russell M., Mudge, Trevor N., and Volz, Richard A., (1986) "Solutions to the n Queens Problem Using Tasking in Ada."., ACM SIGPLAN Notices, 21, 99-110.

[9] Clay, C., (1986) "A New Solution to the N≤8 Queens Problem", ACM SIGPLAN Notices, 21: 28-30.

[10] Cockayne, E. J., and Hedetniemi, S. T., (1986) "On the Diagonal Queens Domination Problem" Journal of Combinatorial Theory, 42, 137-139.

[11] Cull, Paul, and De Curtins, J., (1978) "Knight's Tour Revisited," *Fibonacci Quarterly* 16: 276–285.

[12] Demirors, O. and Tanik, M. M., (1991) "Peaceful Queens and Magic Squares" Technical Report 91-CSE-7, Dept. of Comp. Sci. and Eng., Southern Methodist University.

[13] Erbas, Cengiz, and Tanik, Murat M., (1992) "N-Queens Problem and its Algorithms", Technical Report 91-CSE-8, Dept. of Comp. Sci. and Eng., Southern Methodist University.

[14] Goldsby, Michael E., (1987) "Solving the 'N≤8 Queens' Problem with CSP and Modula", ACM SIGPLAN Notices, 43-52.

[15] Gutiérrez-Naranjo, Miguel A., Et Al. (2009) "Solving The N-Queens Puzzle With P Systems." Seventh Brainstorming Week On Membrane Computing 1: 199-210.

[16] Hoffman, E., J., Loessi, J. C., and Moore, R. C., (1969) "Constructions for the Solution of the m Queens Problem", Mathematics Magazine, XX, 66-72.

[17] Horowitz, E., and Sahni, S., (1979) “Fundamentals of Computer Algorithms”, Computer Science Press, Rockville, Maryland.

[18] Lijo, V. P. and Jasmin T. Jose (2015) “ Solving N-Queen Problem by Prediction” International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 3844-3848

[19] Sosic, Rok, and Gu, Jun, (1990) "A Polynomial Time Algorithm for the N-Queens Problem", SIGART Bulletin 1(3), 7-11.