Generalizing Euler’s Knight’s Tours

Closed Hamiltonian Cycles

Abstract

In this final project for Combinatorics (Fall 2020), we generate and investigate Hamiltonian cycles found in the game of chess. The different graphs generated from the moves of king, queen, bishop, knight, and rook are further studied. We generate functions and other related combinatorial definitions to find integer sequence and better understand the concept of generating functions and Hamiltonian cycles.

Eshaan Gandhi

Soham Palande

Ravali Bommanaboina

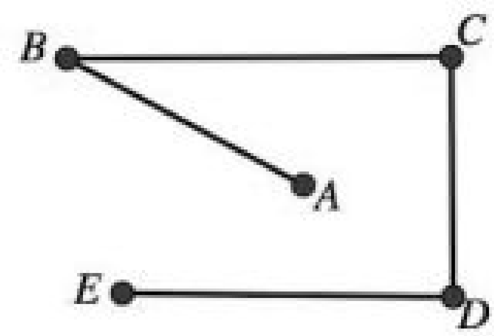
Treasa Bency Biju

Introduction:

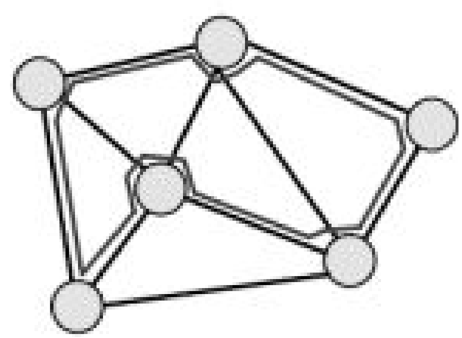
This is the final project for the class Combinatorial Theory taught by Dr. Doron Zeilberger at Rutgers University in the Fall 2020 semester. In this project we seek to extend and generalize the famous Euler's Knight's tours on a chess board. We also explore the integer sequences of closed tours of the different chess pieces and various fictional pieces on k x n boards and the generating functions for such sequences as well. We created a library of functions to be able to construct integer sequences of the total number of closed Hamiltonian paths on chess boards of size k x n for various chess pieces. The integer sequences are generated from the generating functions in the code. We created several "new" chess pieces with "custom" moves and also found their integer sequences. Some of these sequences are already registered with the On-Line Encyclopedia of Integer Sequences® (OEIS®). This document explains the functions and the algorithm we used to find them.

Hamiltonian Paths and Hamiltonian Cycle:

In the mathematical field of graph theory, a Hamiltonian path (or traceable path) is a path in an undirected or directed graph that visits each vertex exactly once. A Hamiltonian cycle (or Hamiltonian circuit) is a Hamiltonian path that is a cycle.



-Hamiltonian Path



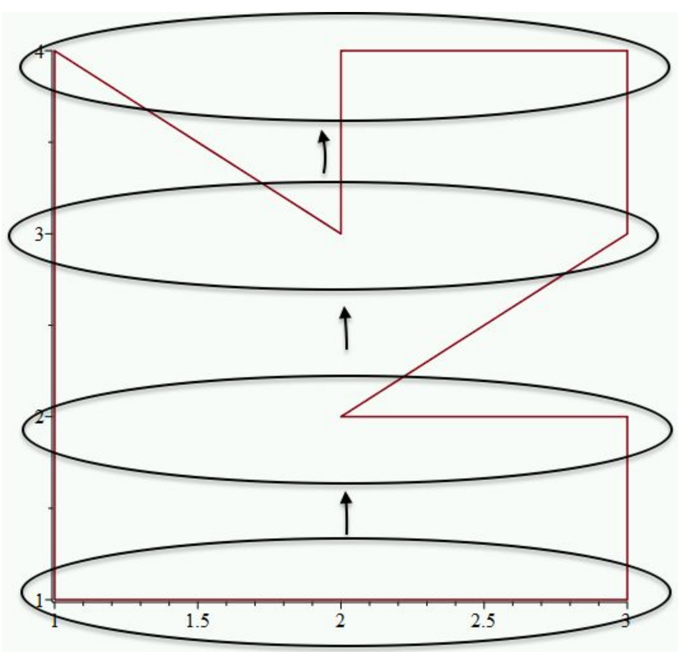
-Hamiltonian Cycle

Methodology :

To find the generating functions for the number of Hamiltonian cycles on k x n boards, we used an **empirical-linguist** approach. We first create a graph of the chess piece for which we want to find a generating function. For each position on a k x n board, this graph "links" all the possible positions to which the given piece can go to in a single move. We then have functions which explore, through brute force, all the possible paths on a given graph which gives the number of Hamiltonian cycles on the given board for a given piece. However, these have an exponential runtime and are impractical for large boards. This motivates us to find generating functions for a given piece on a k x n board where k is fixed. This approach is outlined below.

Consider a k x n board for an arbitrary chess piece.

1. Given a single Hamiltonian cycle of this piece on a k x n board, we use a function (PtoW) which converts the given Hamiltonian cycle to a "word". To illustrate this consider the first column of the k x n board. In the figure below, the first column is represented by the bottommost circle and corresponds to the first column of a k x n board, the second column is represented by the second circle from the bottom and represents the second column in a k x n board. The red line outlines a single Hamiltonian Cycle on a 3 x 4 board. (You can think of the figure below as a chessboard rotated 90 degrees counterclockwise)



2. We define the <u>steps</u> that a piece can take from a given position on the cycle as the number of moves required to reach the “previous” and “next” positions on the Hamiltonian cycle. For example, consider a piece at (1,1) on the board above. So, from (1,1) the piece can reach positions (2,1) and (1,2) on the outlined Hamiltonian path(red). In order to do so, the piece must move (1,0) to reach (2,1) and move (0,1) to reach (1,2). Thus the steps of this piece from the position (1,1) are [(1,0),(0,1)]. Now we consider all the positions that the piece will occupy in the given Hamiltonian cycle that are in the first column. So in the example above, these positions are (2,1) and (3,1). Then we find the “steps” of these positions using the same procedure we did for the position (1,1). For (2,1) the piece can go to either (1,1) or (3,1) and so the steps are (0,-1) and (0,1) respectively. So the steps for position (2,1) are [(0,-1),(0,1)]. Similarly for position (3,1) we get that the steps are [(1,0),(0,-1)].

   The list of steps of all the positions in the first column:
   [[(1,0),(0,1)], [(0,-1),(0,1)], [(1,0),(0,-1)]] make up the first “letter” of the word.

   We then repeat this procedure for all the columns of the board and obtain the respective “letters” to get a “word”. Each Hamiltonian cycle corresponds to a unique word.

3. Then we use a procedure to obtain the entire “language”. This procedure finds the words for every Hamiltonian cycle of the given piece on a k x n board for a specific k and n. The set of all the words is the “language” of that specific k x n board for the given piece.
4. Given the “language” of a board for a given piece, we create a “linguistic graph” of the “words” in the language which gives for each “letter” (or vertex of the graph), the list of letters that can come after it (its neighbors). For example, consider:

   Lang({[s,o,h,a,m]}) would give:
   [{a, h, m, o, s}, {[a, m], [h, a], [o, h], [s, o]}]

   Here, a-> [a,m] since either a or m can appear immediately after a
   Likewise, h->[h,a] since either h or a can appear immediately after h
   For m, there is no set since there are no letters than can appear after m.

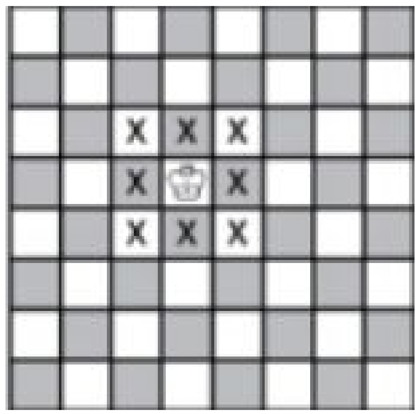
   Putting this into context, the words we are referring to represent the Hamiltonian cycles on a specific k x n board whose letters are the set of steps taken by a piece for all the positions in one column of the k x n board (refer to #2 above)

Thus, we create a linguistic graph and attempt to “learn” or represent the “grammatical rules” of the language for this specific k x n board.

5. We then use a function to empirically “learn” the “entire” grammar of all k x n boards for a fixed k and 1<=n<=max where max is the number of boards (k x max) that we want to test. Our goal is to learn the “entire” grammar for all k x n boards where k is fixed and n=1...infinity. However, there is only a finite number of such grammatical rules for any such k x n board. As soon as the linguistic graph (refer to #6) is identical for consecutive values of n, we have “learned” or acquired all the rules or the “entire grammar” for all k x n boards for a fixed k and n=1...infinity.

6. Finally, in order to find the generating function whose coefficients (in its Taylor series) of t^n is the total`number of Hamiltonian cycles on a k x n board, we try to “learn” the entire grammar by empirically trying different values of max (refer to #5). As soon as consecutive values of n (for n=1...max) generate equivalent linguistic graphs for a k x n board, the process converges and returns the latest linguistic graph. Then, we use another procedure to find the weighted enumerator according to the weight t^LengthOfWalk, of the set of paths in the linguistic graph and then generate the corresponding generating function.

Chess piece – KING:



King's moves

The king is the most important piece in the game of chess. If a player's king is threatened with capture, it is said to be in check, and the player must remove the threat of capture on the next move. If this cannot be done, the king is said to be in checkmate, resulting in a loss for that player.

The moves of a king are shown above and the MAPLE function uses this basic idea to establish generating functions, graphs and Hamiltonian cycles of those graphs.

Main functions for KING:

- A graphical representation of the King's moves on a (3 x 2) board

  KiG(3, 2) := [{2, 3, 4}, {1, 3, 4}, {1, 2, 4, 5, 6}, {1, 2, 3, 5, 6}, {3, 4, 6}, {3, 4, 5}], [[1, 1], [1, 2], [2, 1], [2, 2], [3, 1], [3, 2]]

- The possible Hamiltonian cycles on a (3 x 2) board

  KiTours(3, 2) := {[[1, 1], [1, 2], [2, 1], [3, 1], [3, 2], [2, 2], [1, 1]], [[1, 1], [1, 2], [2, 1], [3, 2], [3, 1], [2, 2], [1, 1]], [[1, 1], [1, 2], [2, 2], [3, 1], [3, 2], [2, 1], [1, 1]], [[1, 1], [1, 2], [2, 2], [3, 2], [3, 1], [2, 1], [1, 1]], [[1, 1], [2, 1], [3, 1], [3, 2], [2, 2], [1, 2], [1, 1]], [[1, 1], [2, 1], [3, 2], [3, 1], [2, 2], [1, 2], [1, 1]], [[1, 1], [2, 2], [3, 1], [3, 2], [2, 1], [1, 2], [1, 1]], [[1, 1], [2, 2], [3, 2], [3, 1], [2, 1], [1, 2], [1, 1]]

- The generating function for King’s tours on a 2 x n board:

  KingToursGF(2, 7, t) :=

  t^2*(2*t - 3)/(2*t - 1)

- First 20 terms of the King’s tours on a (3 x i) chessboard

  seq(nops(KiTours(3, i))/2, i = 1 .. 20)

  0, 4, 16, 120, 744, 4922, 31904, 208118, 1354872, 8826022, 57483536, 374412158, 2438639080, 15883563110, 103454037120, 673825180718, 4388811619032, 28585557862518, 186185731404016, 1212679737590398

- First 20 terms of the King’s tours on a (2 x i) chessboard

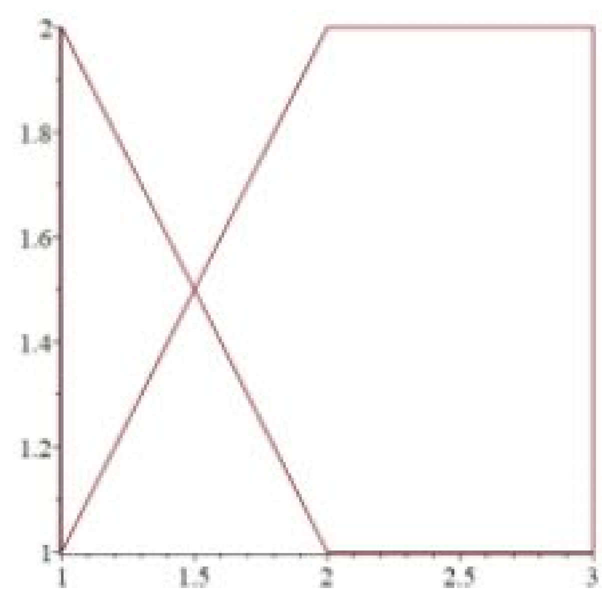
  seq(nops(KiTours(2, i))/2, i = 2 .. 20

  3, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048

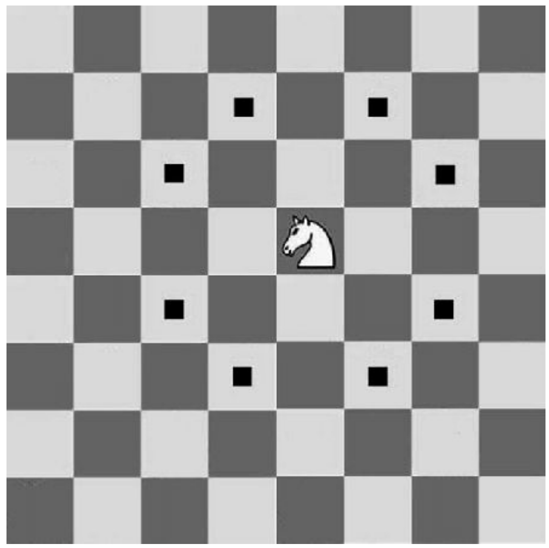
  OEIS A-number := A198633

- Graph of a single KING’s tour on MAPLE

  plot(KiTours(3, 2)[1])



Chess piece – KNIGHT:



Knight's moves

The knight is a piece in the game of chess and is represented by a horse's head and neck. Each player starts with two knights, which are located between the rooks and bishops in the standard starting position.

The knight's moves are shown above. The graphical representation, Hamiltonian cycles and generating functions for the knight are worked on MAPLE.

Main Functions for KNIGHT:

- A graphical representation of the moves of a knight on a (3 x 2) chessboard.

  KtG(3, 2) := [{6}, {5}, {}, {}, {2}, {1}], [[1, 1], [1, 2], [2, 1], [2, 2], [3, 1], [3, 2]]

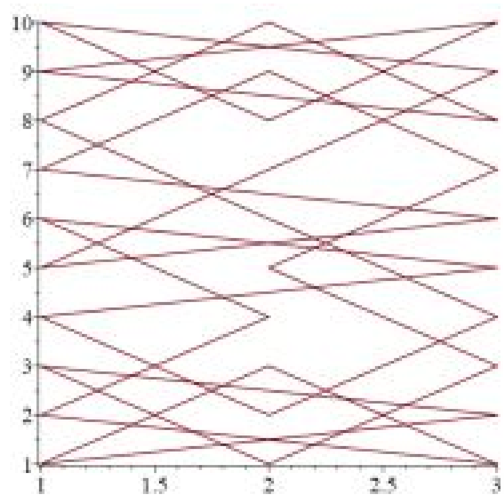
- The Hamiltonian cycles made by the knight's moves on a (3 x 2) chessboard. The result is divided by 2 to get the correct answer, since the path considers to – from logic.

  nops(KtTours(3, 10))/2 := 16

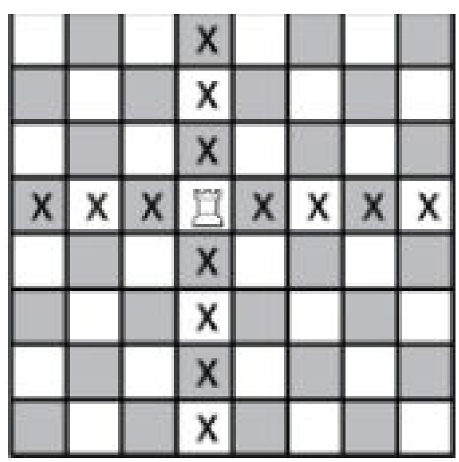
- First 30 terms of Knight Tours on a (3 x i) chessboard:
  seq(nops(KtTours(3, i)), i = 1 .. 30)
  0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 352, 0, 3072, 0, 30848, 0, 295456, 0, 2896832, 0, 28120096, 0, 273895232, 0, 2664515712, 0, 25931157504, 0, 252338724352
- First 15 terms of Knight's Tours on a (2 x i) chessboard are all 0.

- Graph for knight's tour on MAPLE

plot(KtTours(3, 10)[1])



Chess piece – ROOK:



Rook's moves

The rook is a piece in the game of chess resembling a castle. Formerly the piece was called the tower, marquess, rector, and comes. The term castle is informal, incorrect, or old-fashioned. Each player starts the game with two rooks, one on each of the corner squares on their own side of the board.
The rook's moves are shown above. Using MAPLE we've found some interesting new functions to study the rook's moves with a combinatorial and graphical approach.

Main Function for a Rook:

- The possible Hamiltonian paths found from a Rook's moves on a (3 x 2) chess board

  RiTours(3, 2);

  {[[1, 1], [1, 2], [2, 2], [3, 2], [3, 1], [2, 1], [1, 1]], [[1, 1], [1, 2], [3, 2], [2, 2], [2, 1], [3, 1], [1, 1]], [[1, 1], [2, 1], [2, 2], [1, 2], [3, 2], [3, 1], [1, 1]], [[1, 1], [2, 1], [3, 1], [3, 2], [2, 2], [1, 2], [1, 1]], [[1, 1], [3, 1], [2, 1], [2, 2], [3, 2], [1, 2], [1, 1]], [[1, 1], [3, 1], [3, 2], [1, 2], [2, 2], [2, 1], [1, 1]]}

- First 7 terms of Rook's Tours on a (3 x i) chessboard

  seq(nops(RiTours(3, i)), i = 1 .. 7)

  0, 6, 96, 3132, 84192, 1821912, 37359444

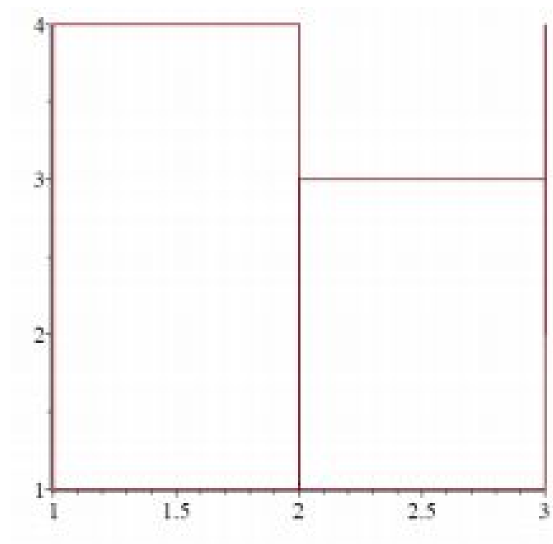
- First 10 terms of Rook's Tours on a (2 x i) chessboard

  seq(nops(RiTours(2, i)), i = 1 .. 10)

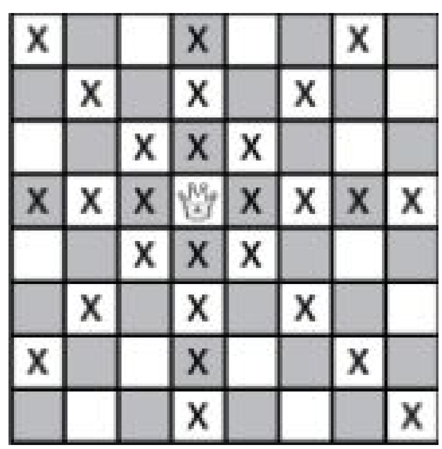
  1, 2, 6, 20, 48, 126, 348, 936, 2512, 6788

- Graph for Rook's tour on MAPLE

  plot(RiTours(3, 4)[1])



Chess piece – QUEEN:



Queen's moves

The queen is the most powerful piece in the game of chess, able to move any number of squares vertically, horizontally or diagonally. Each player starts the game with one queen, placed in the middle of the first rank next to the king. Because the queen is the strongest piece, a pawn is promoted to a queen in the vast majority of cases.

The moves of a queen are shown above. We further study the Queen's Hamiltonian cycles on different graphs below.

Main functions of the Queen:

- The graph of a queen’s move on a (3 x 1) chessboard

  QiG(3,1)

  {[-3, -3], [-3, 3], [-2, -2], [-2, 2], [-1, -1], [-1, 0], [-1, 1], [0, -3], [0, -2], [0, -1], [0, 1], [0, 2], [0, 3], [1, -1], [1, 0], [1, 1], [2, -2], [2, 2], [3, -3], [3, 3]} {[-3, -3], [-3, 3], [-2, -2], [-2, 2], [-1, -1], [-1, 0], [-1, 1], [0, -3], [0, -2], [0, -1], [0, 1], [0, 2], [0, 3], [1, -1], [1, 0], [1, 1], [2, -2], [2, 2], [3, -3], [3, 3]} {[-3, -3], [-3, 3], [-2, -2], [-2, 2], [-1, -1], [-1, 0], [-1, 1], [0, -3], [0, -2], [0, -1], [0, 1], [0, 2], [0, 3], [1, -1], [1, 0], [1, 1], [2, -2], [2, 2], [3, -3], [3, 3]} [{2}, {1, 3}, {2}], [[1, 1], [2, 1], [3, 1]]

- The graph of a queen’s move on a (2 x 1) chessboard

  QiG(2, 1);

  {[-2, -2], [-2, 2], [-1, -1], [-1, 0], [-1, 1], [0, -2], [0, -1], [0, 1], [0, 2], [1, -1], [1, 0], [1, 1], [2, -2], [2, 2]} {[-2, -2], [-2, 2], [-1, -1], [-1, 0], [-1, 1], [0, -2], [0, -1], [0, 1], [0, 2], [1, -1], [1, 0], [1, 1], [2, -2], [2, 2]} [{2}, {1}], [[1, 1], [2, 1]]
- The Hamiltonian cycles found in a Queen’s graph on a (3 x i) chessboard

  seq(SAWnu(QiG(3, i)[1])/2, i = 1 .. 4)

  0, 24, 1960, 243040

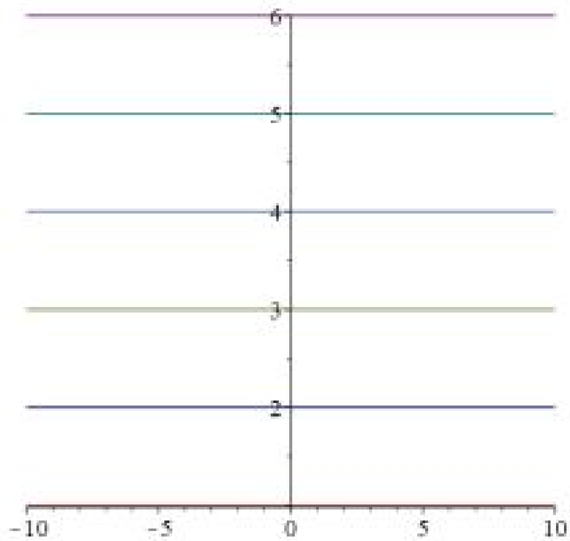
- The Hamiltonian cycles found in a Queen’s graph on a (2 x i) chessboard

  seq(SAWnu(QiG(2, i)[1])/2, i = 1 .. 9)

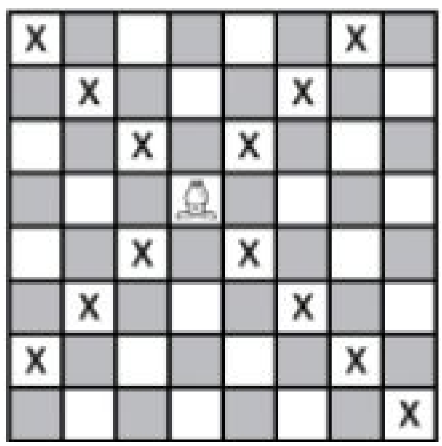
  1, 3, 24, 108, 522, 2646, 13150, 65206, 324370

- Graph of Queen’s tour on a (3 x 2) chessboard

  plot(SAW(QiG(3, 2))[1])



Chess piece – BISHOP:



Bishop's move

The bishop is a piece in the game of chess. Each player begins the game with two bishops. One starts between the king's knight and the king, the other between the queen's knight and the queen. The starting squares are c1 and f1 for White's bishops, and c8 and f8 for Black's bishops.

The above picture shows the moves that are legal for a bishop. We further generated sequences through MAPLE to better understand the combinatorial and graphical interpretation behind this.

Main functions for a Bishop:

• The Bishop's move graph on a (3 x 2) chessboard

BiG(3, 2);

[{4}, {3}, {2, 6}, {1, 5}, {4}, {3}], [[1, 1], [1, 2], [2, 1], [2, 2], [3, 1], [3, 2]]

• The Bishop's move graph on a (2 x 2) chessboard

BiG(2, 2);

[{4}, {3}, {2}, {1}], [[1, 1], [1, 2], [2, 1], [2, 2]

• The Bishop's move is unique. Hamiltonian cycles are not found in Bishop graphs. Since Bishops can only move on one color, Hamiltonian cycles are impossible unless we modify the Bishop's moves.