Greedy Chess Evaluation on Forking Position

Implementation of Greedy Algorithm to Capture or Defend Chess Piece Due to Forking

Steven - 13520131
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): stevenwen81@gmail.com

***Abstract*—Chess is a board game that has lived through the centuries. There has been a lot of great chess players over the years. All of them possessed great analytical skills regarding chess. However, for beginner players they do not have such skills and as such they require helps. We would like to help them with an algorithm and show them the best move in chess forking position. In the chess game, each of the pieces has already been assigned its own worth and as such we are able to use the greedy algorithm to calculate what to do based on worth of the pieces.**

Keywords—Chess, Decision, Chess Forking Position, Greedy Algorithm

I. Introduction

Chess is a board game that has 8x8 grid which can be played between two players (mostly represented with the color White and Black). The two players will take turn to play according their strategy.

Figure 1. Chess Board

Source:

https://images.chesscomfiles.com/uploads/v1/images_users/tiny_mce/PedroPinhata/phpZTvydV.png

At the beginning of the game, chess board contains 32 pieces and of those pieces there are 6 distinct types: King, Queen, Bishop, Knight, Rook, and Pawn. Each of the chess piece are located in a different place. Normally, the location of a piece can be called by XY in which X represent the alphabetical order on the board and Y represent the numerical order on the board.

In the game of chess, the king represents the player. It is due to the fact that when the king is check mated, the player immediately loses.

At the start of a chess game:

- Each player has eight pieces of pawn. The White pawn is located on the second rank of the board while the Black pawn is located on the seventh rank of the board. The pawn can only move forward. If it is the pawn first move, it can move forward one or two squares forward. Else, it can only move one square forward. The pawn can only capture opponent's piece one square of it's left or right diagonal.
- Each player has two pieces of bishop. The White bishop is located on c1 and f1 while Black bishop is located on c8 and f8. The bishop can move multiple squares on it's diagonal as long as it is not blocked by any pieces. The bishop can capture opponent's piece by moving diagonally to the square which is occupied by the other player's piece.
- Each player has two pieces of knight. The White knight is located on b1 and g1 while Black knight is located on b8 and g8. The knight can move one square left or right horizontally and then moves two squares up or down vertically and also it can move two squares left of right horizontally and then move one square up or down vertically. While moving, the knight can jump over other pieces which mean the knight cannot be blocked by another pieces. The knight can capture opponent's pieces by moving to square which is occupied by the other player's piece.
- Each player has two pieces of rook. The White rook is located on a1 and h1 while Black rook is located on a8 and h8. The rook can move multiple squares vertically or horizontally as long as it is not blocked by any

pieces. The rook can capture opponent's piece by moving horizontally or vertically to the square which is occupied by the other player's piece.

- Each player has one piece of queen. The White queen is located on d1 while the Black queen is located on d8. The queen movement and capture mechanism is the combination of both bishop and rook.
- Each player has one piece of king. The White king is located on e1 while Black King is located on e8. The king can only move one square to any direction. The king can capture opponent's piece by moving to any direction where the square which is occupied by the other player's piece.

II. THERORETICAL FOUNDATION

A. *Greedy Algorithm*

Greedy Algorithm is the most popular and simple algorithm to solve optimization problem such as maximization and minimization. Greedy algorithm solves a problem by taking the best it can take in each step of solving which the decision in each step of the solving is hoping to obtain the global optima.

There are some elements that makes up the Greedy Algorithm.

- *Candidate Set*

  The candidate set contains the candidate that will be chosen each step for the greedy solution. Which later on a candidate will be chosen from this set using the selection function.

- *Solution Set*

  The solution set contains the chosen candidate. Which will be checked using the solution function.

- *Solution Function*

  The solution function determines whether or not the chosen candidate has provided us with solution.

- *Selection Function*

  The selection function chose a candidate based on the greedy algorithm strategy.

- *Feasibility Function*

  The feasible function will check whether or not the chosen candidate is eligible or not to be inserted into solution set.

- *Objective Function*

  The objective function refers to the overall goal of the optimization problem (maximization of minimization).

B. *Chess Piece Worth*

In chess, each of the piece has it's own worth. For example, the pawn is the least powerful piece, the bishop and the knight is a minor piece, the rook and the queen is the major piece. This then leads to the numerical worth which are shown in the table below.

Piece Type	Numerical Worth
Pawn	1
Bishop	3
Knight	3
Rook	5
Queen	9

Table 1. Chess Piece Worth

Source:

https://www.chess.com/terms/chess-piece-value

From the table above, it could be seen that each type of piece has it's own numerical worth. From this numerical worth, we will then use it to apply Greedy Algorithm.

C. *Forking Position*

In chess, there are so many tactics. One of the famous tactics which are used is called fork. A position is called a forking position whenever there is a piece that attacks two or more of the opponent's pieces at the same time. There are so many chess forking positions. Some of the forking positions are shown below:

- Pawn Fork

  A pawn fork position happens when a pawn attack 2 pieces at the same time.

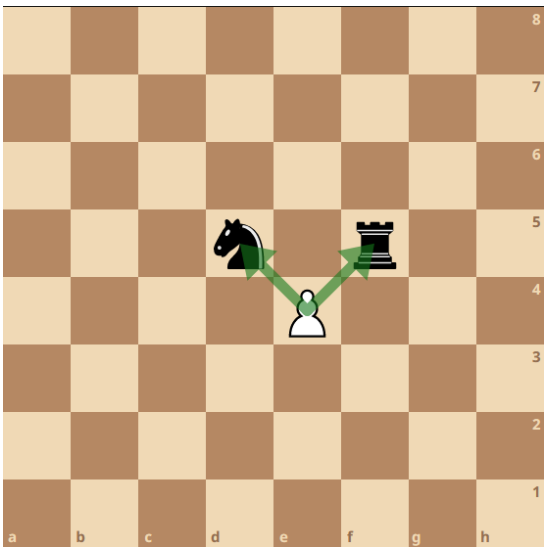

Figure 2. Pawn Fork

Source:

https://lichess.org/editor/8/8/8/3n1r2/4P3/8/8/8_w_-_-_0_1

- Bishop Fork

  A bishop fork position happens when a bishop attack 2 or more pieces at the same time.

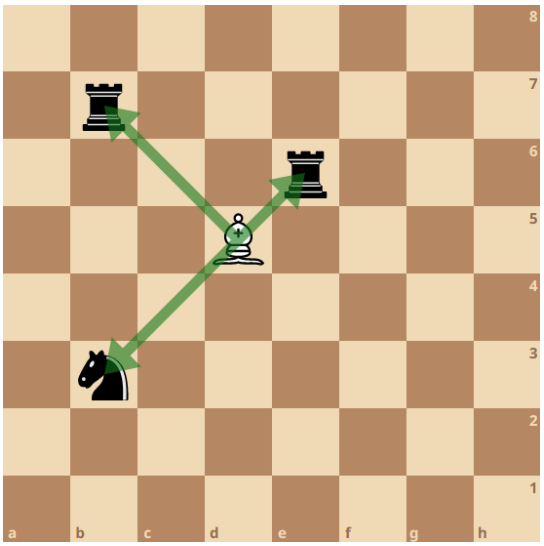

Figure 3. Bishop Fork

Source:

https://lichess.org/editor/8/1r6/4r3/3B4/8/1n6/8/8_w_-_-_0_1

- Knight Fork

  A knight fork position happens when a knight attack 2 or more pieces at the same time.

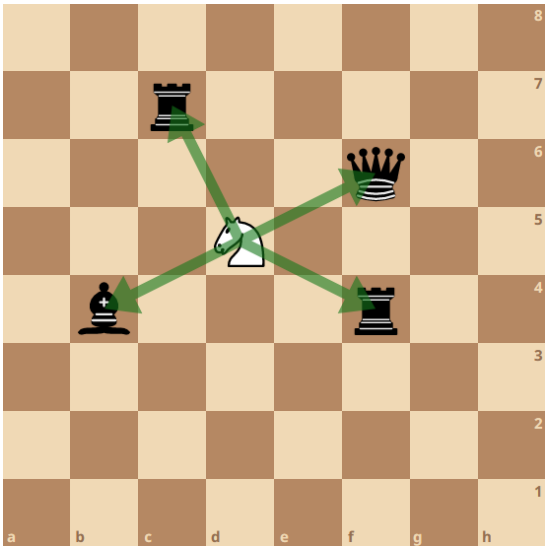

Figure 4. Knight Fork

Source:

https://lichess.org/editor/8/2r5/5q2/3N4/1b3r2/8-----/8/8_w_-_-_0_1

- Rook Fork

  A rook fork position happens when a rook attack 2 or more pieces at the same time.

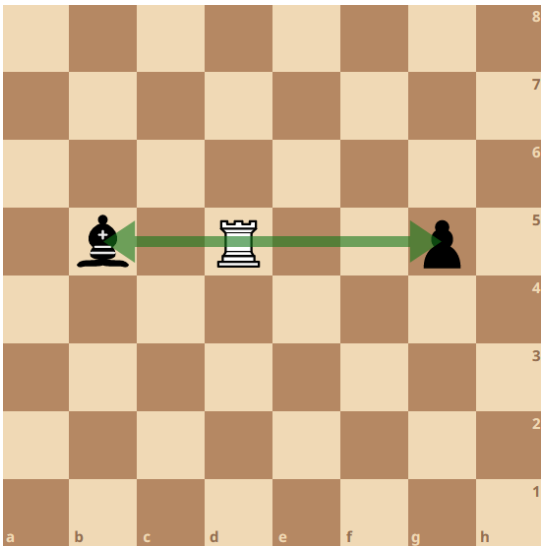

Figure 5. Rook Fork

Source:

https://lichess.org/editor/8/8/8/1b1R2p1/8/8/8/8_w_-_-_0_1

- Queen Fork

  A queen fork position happens when a queen attack 2 or more pieces at the same time.

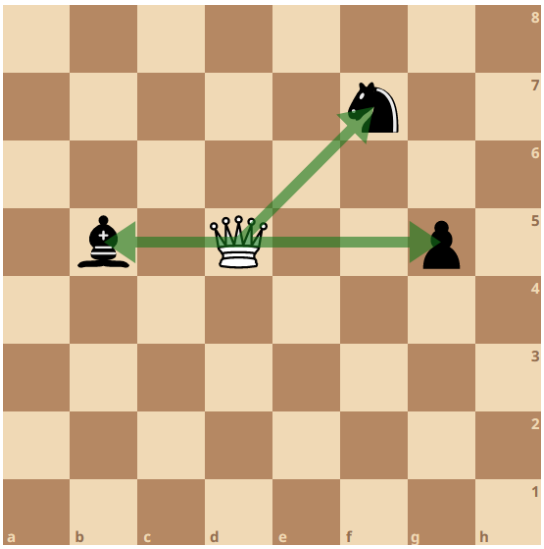

Figure 6. Queen Fork

Source:

https://lichess.org/editor/8/5n2/8/1b1Q2p1/8/8/8/8_w_-_-_0_1

- King Fork

  A king fork position happens when a king attack 2 or more pieces at the same time.

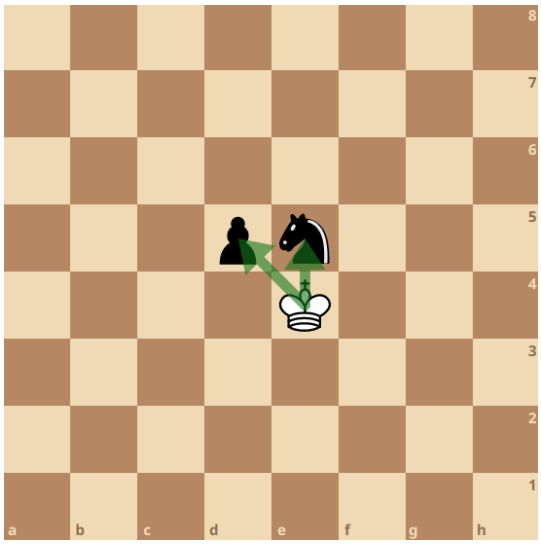

Figure 7. King Fork

Source:

https://lichess.org/editor/8/8/8/3pn3/4K3/8/8/8_w_-_-_0_1

III. Greedy Algorithm Implementation in Forking Position

A. *Attacking Perspective*

- Pawn Fork

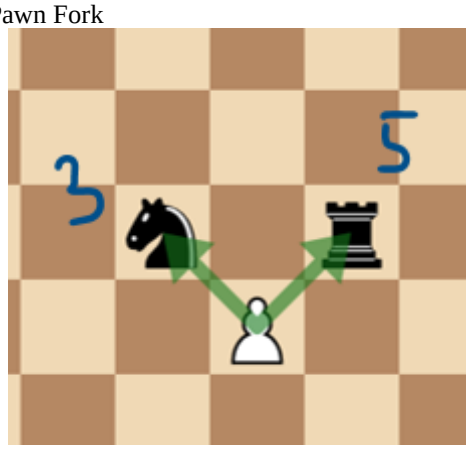

Figure 8. Pawn Fork Attacking Evaluation

From the image above, we could see that the pawn is forking a knight and a rook. From *Table 1*, we know that knight has a worth of 3 and rook has a worth of 5. Therefore, the best move White can do according to the Greedy Algorithm is using the pawn to capture the opponent's rook.

- Bishop Fork

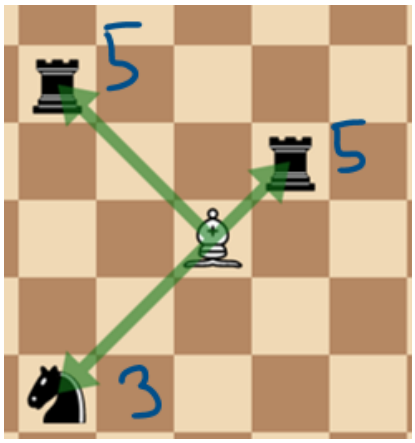

Figure 9. Bishop Fork Attacking Evaluation

From the image above, we could see that the bishop is forking two rooks and a knight. From *Table 1*, we know that knight has a worth of 3 and rook has a worth of 5. Therefore, the best move White can do according to the Greedy Algorithm is using the bishop to capture the opponent's rook (either rook is fine because both has the same worth of 5).

- Knight Fork

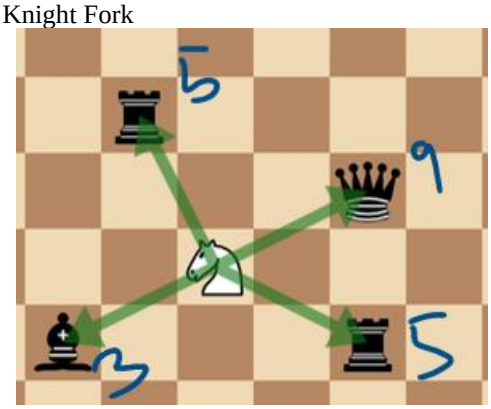

Figure 10. Knight Fork Attacking Evaluation

From the image above, we could see that the knight is forking two rooks, a bishop, and a queen. From *Table 1*, we know that bishop has a worth of 3, rook has a worth of 5, and queen has a worth of 9. Therefore, the best move White can do according to the Greedy Algorithm is using the knight to capture the opponent's queen.

- Rook Fork

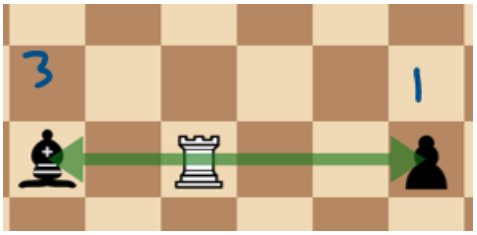

Figure 11. Rook Fork Attacking Evaluation

From the image above, we could see that the rook is forking a pawn and a bishop. From *Table 1*, we know that pawn has a worth of 1 and bishop has a worth of 3.

Therefore, the best move White can do according to the Greedy Algorithm is using the rook to capture the opponent's bishop.

- Queen Fork

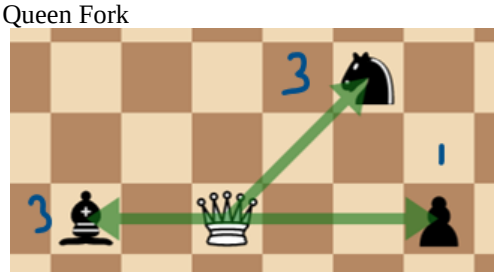

Figure 12. Queen Fork Attacking Evaluation

From the image above, we could see that the queen is forking a pawn, a bishop, and a knight. From *Table 1*, we know that pawn has a worth of 1, bishop has a worth of 3, and knight has the worth of 3. Therefore, the best move White can do according to the Greedy Algorithm is using the queen to capture either the opponent's bishop or the opponent's knight (either bishop or knight is fine because both has the same worth of 3).

- King Fork

Figure 13. King Fork Attacking Evaluation

From the image above, we could see that the king is forking a pawn and a knight. From *Table 1*, we know that pawn has a worth of 1 and knight has a worth of 3. Therefore, the best move White can do according to the Greedy Algorithm is using the king to capture the opponent's knight.

B. Defending Perspective

- Pawn Fork

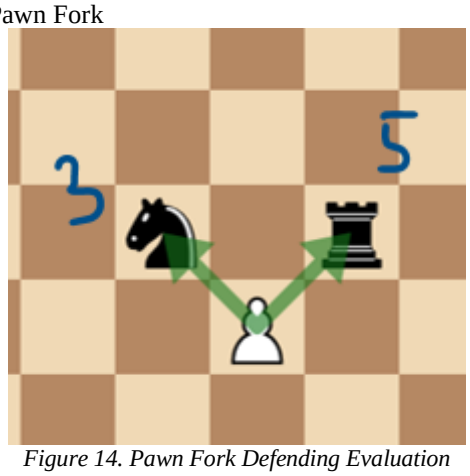

Figure 14. Pawn Fork Defending Evaluation

From the image above, we could see that the pawn is forking a knight and a rook. From *Table 1*, we know that knight has a worth of 3 and rook has a worth of 5. Therefore, the best move Black can do according to the Greedy Algorithm is to safe it's rook and sacrifice it's knight.

- Bishop Fork

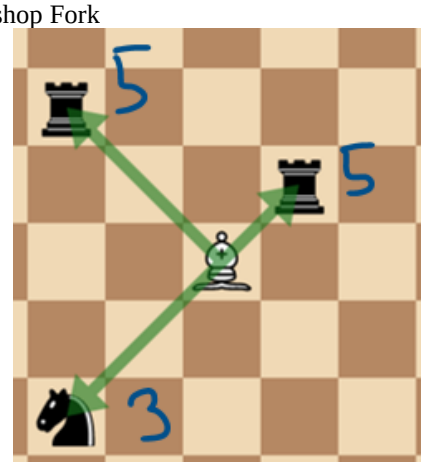

Figure 15. Bishop Fork Defending Evaluation

From the image above, we could see that the bishop is forking two rooks and a knight. From *Table 1*, we know that knight has a worth of 3 and rook has a worth of 5. Therefore, the best move Black can do according to the Greedy Algorithm is to save one of it's rook and sacrifice either it's knight or it's rook.

- Knight Fork

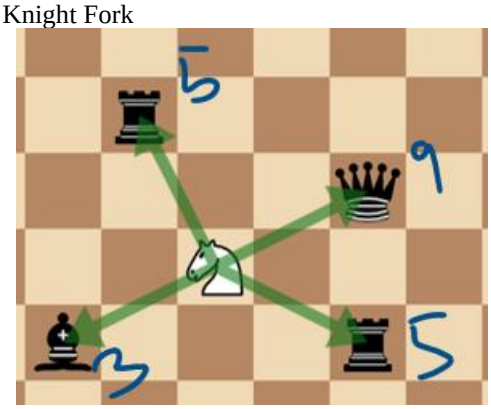

Figure 16. Knight Fork Defending Evaluation

From the image above, we could see that the knight is forking two rooks, a bishop, and a queen. From *Table 1*, we know that bishop has a worth of 3, rook has a worth of 5, and queen has a worth of 9. Therefore, the best move Black can do according to the Greedy Algorithm is to save it's queen and sacrifice either it's bishop or one of it's rooks.

- Rook Fork

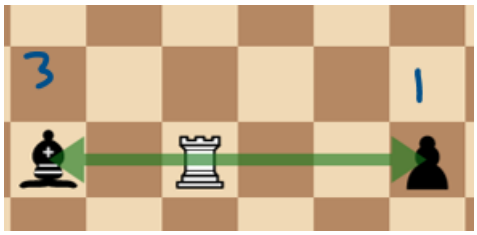

Figure 17. Rook Fork Defending Evaluation

From the image above, we could see that the rook is forking a pawn and a bishop. From *Table 1*, we know that pawn has a worth of 1 and bishop has a worth of 3. Therefore, the best move Black can do according to the Greedy Algorithm is to save it's bishop and sacrifice it's pawn.

- Queen Fork

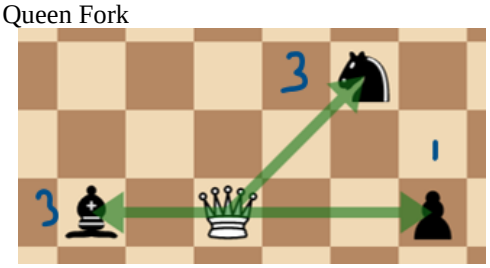

Figure 18. Queen Fork Defending Evaluation

From the image above, we could see that the queen is forking a pawn, a bishop, and a knight. From *Table 1*, we know that pawn has a worth of 1, bishop has a worth of 3, and knight has a worth of 3. Therefore, the best move Black can do according to the Greedy Algorithm is to save either it's bishop or it's knight and sacrifice one of the rest.

- King Fork

Figure 19. King Fork Defending Evaluation

From the image above, we could see that the king is forking a pawn and a knight. From *Table 1*, we know that pawn has a worth of 1 and knight has a worth of 3. Therefore, the best move Black can do according to the Greedy Algorithm is to save it's knight and sacrifice it's pawn.

C. *Greedy Elements*

- Candidate Set

  All the pieces that are currently on the forking position will be treated as the candidate set. This is because it is the objective of this case which is a player decision on capturing or defending pieces from a fork.

- Solution Set

  The solution set shall refer to what piece(s) that will be played on certain round which means that each round might have different solution set.

- Solution Function

  The solution function in this case is all available forking position piece(s) moves.

- Selection Function

  To implement greedy algorithm in this forking position, we will first assign a numerical value to each piece that are at the forking position for example queen is 9, rook is 5, knight is 3, etc.

- Feasible Function

  The feasible function in this case is a function that will check if or whether the player can do that move or not. As example when player A wants to capture player B's knight via forking but player A's king in is a check, thus player A have to secure his king first.

- Objective Function

  The objective function in this case is divided by two. The first one is when we are on a forking position and we are the attacker which means we are going to capture the opponent's piece. In this case, the objective function is defined as the piece with the highest value. The second is when we are on a forking position and we are the defender which means we are going to secure our most valuable pieces that are currently being forked by our opponent's. In this case, the objective function is also defined as the piece with the highest value.

IV. CONCLUSION

Greedy algorithm is an algorithm that is widely used because of its easy implementation in an optimization problem and one of the implementations could be seen in this paper where the greedy algorithm is used to make an optimum result at a forking position. Even though the greedy algorithm is not the best algorithm to get the best solution on the forking position, however, this algorithm still stood a chance in most of the forking position cases.

In Conclusion, this strategy is still far from being a decent strategy, and as such, it still needs an upgrade or rework by combining it with another algorithm. This is due to the greedy characteristic which is taking the best that it can take now without consideration for what comes afterwards, and not caring whether or not the current decision taken will result in either a loss or gain in the future.

VIDEO LINK AT YOUTUBE

The following link leads to a video of an explanation about this strategy.

https://youtu.be/QYb2dbbYfXk

ACKNOWLEDGMENT

The author would like to thank Dr. Nur Ulfa Maulidevi, S.T, M.Sc. as the author's lecturer for the Algorithm Strategy course, as well as other lecturers for the past guidance and knowledge. The author would also like to thank his family and friends for the supports given.

REFERENCES

[1] Chess.com, "Chess Pieces", https://www.chess.com/terms/chess#what-is-chess, accessed 3 May 2022

[2] Chess.com, "Chess Pieces Value", https://www.chess.com/terms/chess-piece-value, accessed 3 May 2022

[3] Chess.com, "What Is Chess", https://www.chess.com/terms/chess#what-is-chess, accessed 3 May 2022

[4] R. Munir, "Algoritma Greedy (Bagian 1), https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf, accessed 3 May 2022

[5] Y. Markushin, "Top 5 Types of Forks Every Chess Player Must Know", https://thechessworld.com/articles/general-information/5-forks-every-chess-player-must-know/, accessed 3 May 2022

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Mei 2022

Steven 13520131