Alpha\_Cv1

Nguyen Anh Minh

Copyright ©2024 Nguyen Anh Minh

Alpha\_Cv1

Alpha\_Cv1 is an AI, which can play Gomoku. Alpha\_Cv1 used the Minimax algorithm and used Alpha Beta Pruning Which makes the Minimax algorithm more optimal.

# CONTENTS

## INTRODUCE

* Why use AI to play Gomoku but don’t use a normal algorithm?
* Why use the Minimax algorithm?

### MINIMAX ALGORITHM

* What is Minimax algorithm?
* How it works?
* How to code?

#### ALPHA BETA PRUNING

* What is Alpha Beta Pruning?
* How it works?
* How to code?

##### **CONSTRUCTION OF ALPHA\_CV1**

* Construction. How it works?
* Training.
* Selection process.
* Choose the best data.
* How to code?

##### **TEST ALPHA\_CV1**

* Effective.
* Strong point.
* Weak point.
* Can it be improved it?

1. **INTRODUCE**

* Alpha\_Cv1 is an AI and it used the Minimax algorithm and used Alpha Beta Pruning which makes the Minimax algorithm more optimal.
  1. Why use AI to play Gomoku but don’t use a normal algorithm?
* Gomoku is an abstract strategy board game. It is played with Go pieces (X and O) on a 19 x 19 board or more. If I use a normal algorithm, I can’t have a high win percentage, and the time to find a way to play is very long. Therefore, using AI is the best way to play with a high win percentage and the time to find a way to play will be reduced.
  1. Why use the Minimax algorithm?
* The Minimax algorithm is an algorithm that can help find the most optimal way to play. It can increase the win percentage of AI. However, if we don’t use it effectively, The Minimax algorithm will make the program need more time to find a way to play. Therefore, we will use Alpha Beta Pruning, which can make the Minimax algorithm more optimal and reduce time.

1. **MINIMAX ALGORITHM**
   1. What is Minimax algorithm?

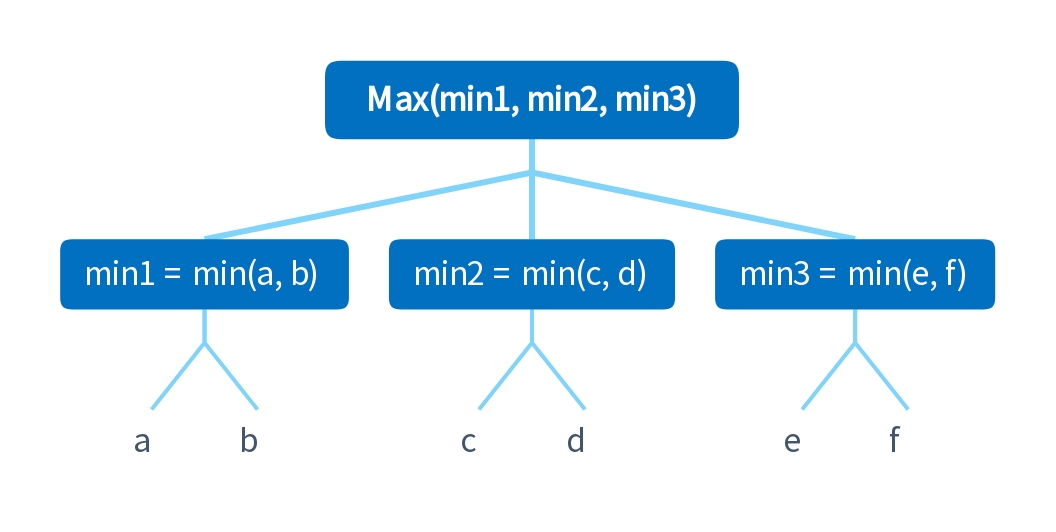
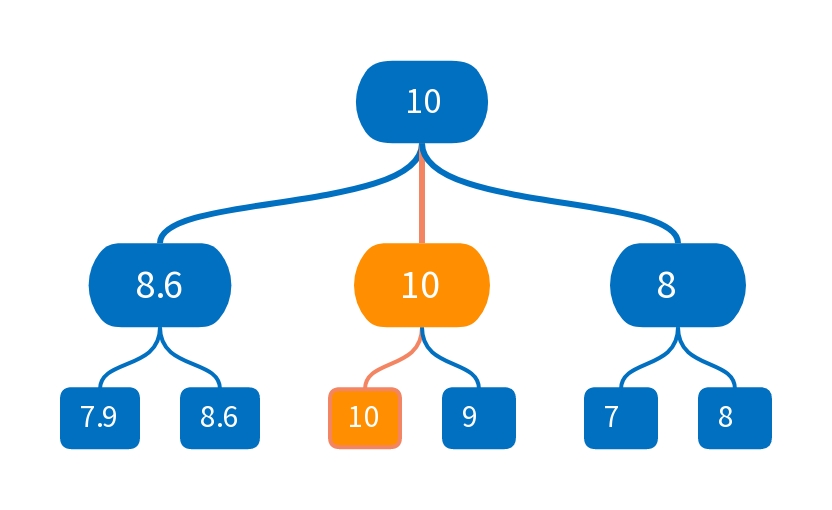
* The Minimax algorithm is a decision rule used in artificial intelligence, decision theory, game theory, statistics, and philosophy for minimizing the possible loss for a worst case (maximum loss) scenario. When dealing with gains, it is referred to as "Maximin" – to maximize the minimum gain. Originally formulated for several-player zero-sum game theory, covering both the cases where players take alternate moves and those where they make simultaneous moves, it has also been extended to more complex games and to general decision-making in the presence of uncertainty.
* In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible. Minimax can use on the tree or use backtracking.
  1.  How it works?

Figure 1. *Minimax algorithm works*

* In one game, we will have a lot of cases to play and we won’t be able to know the best case which helps us win. Therefore, we will use a tree (Figure 1) to find all cases happening and each child node will be a case happening.
* We use the root node as the first case of player one and any nodes in odd number level are the case of player two. Each node in any level contains 1 integer (point of that case) which gets the min or max of which integer from the child node of each node there. (Figure 1).
* The integer of the root node is the highest point of which one case we can get. From that point, we can find the best case we can play. Example
  1. How to code?
* We will build a tree by Object-oriented programming (OOP)

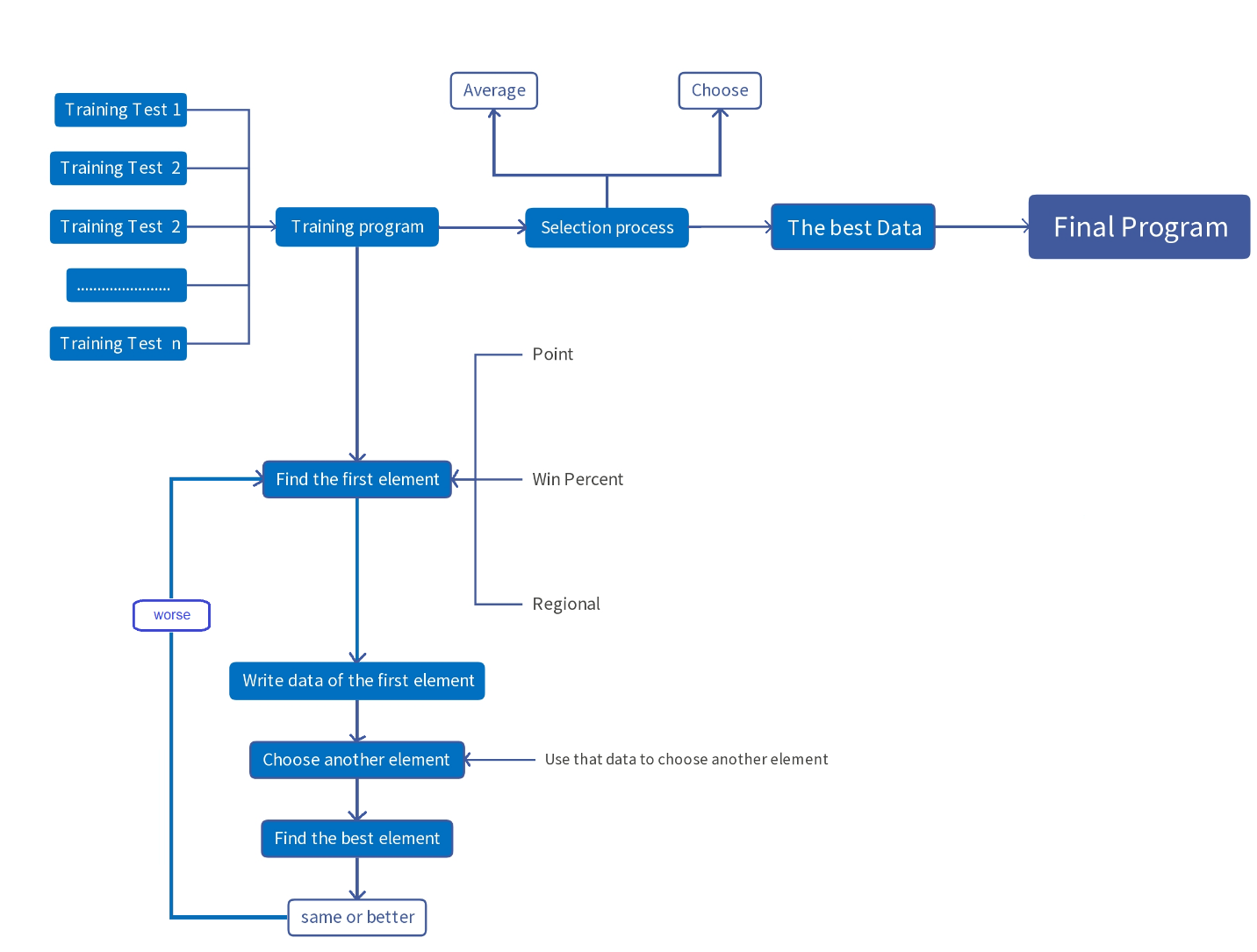
1. **ALPHA BETA PRUNING**
2. **CONSTRUCTION OF ALPHA\_CV1**

Figure 1.

1. **TEST ALPHA\_CV1**

Figures title:

Figure 1.

For more information about all elements of APA formatting, please consult the APA Style Manual, 6th Edition.