

## Breve introducción al algoritmo Minimax:

### Minimax:

La idea es crear un juego de Ta Te Ti invencible, para ello es necesario calcular todas las posibles jugadas y usar algo para determinar el movimiento óptimo.

### Describiendo el juego perfecto de Ta Te Ti.

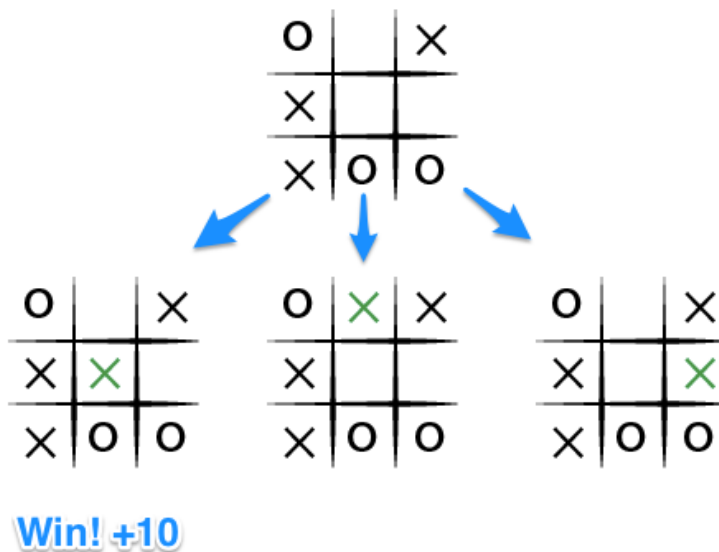
Jugar un juego perfecto significa en el peor de los casos empatar. Si juego con otro jugador perfecto el juego siempre terminará en empate.

Vamos a asignar un puntaje a la condición de final de juego

- Gano: 10 puntos
- Pierdo: -10 puntos
- Empato: 0 puntos

### Ejemplo:

Tomemos el ejemplo de un juego cerca del final, donde es turno del jugador (X). La meta es maximizar el puntaje final.

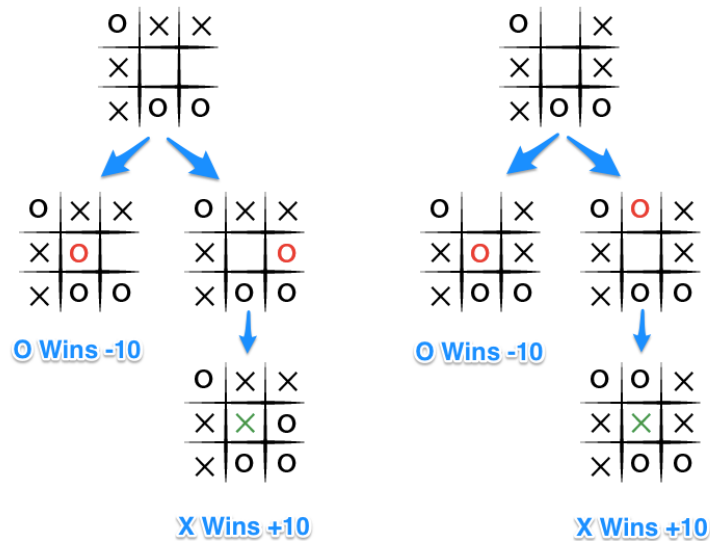


Hay 3 opciones de movimiento que el jugador X tiene. Una claramente lo consagra ganador entonces obtiene 10 puntos.

Si no se hacen esos movimientos el jugador O podría ganar fácilmente. La meta de X es escoger el movimiento de máximo puntaje.

### Acerca del player O:

Asumimos que el player O también está jugando para ganar el juego, pero el jugador O lo que quiere es elegir un movimiento que resulte en el peor puntaje para el jugador X, quiere elegir un movimiento que minimice el último puntaje. Miremos desde la perspectiva del jugador O, empezando con los dos juegos anteriores en los cuales X gana inmediatamente.



El jugador O elegirá un movimiento que resulte en -10 para el jugador X.

### Describiendo el algoritmo Minimax:

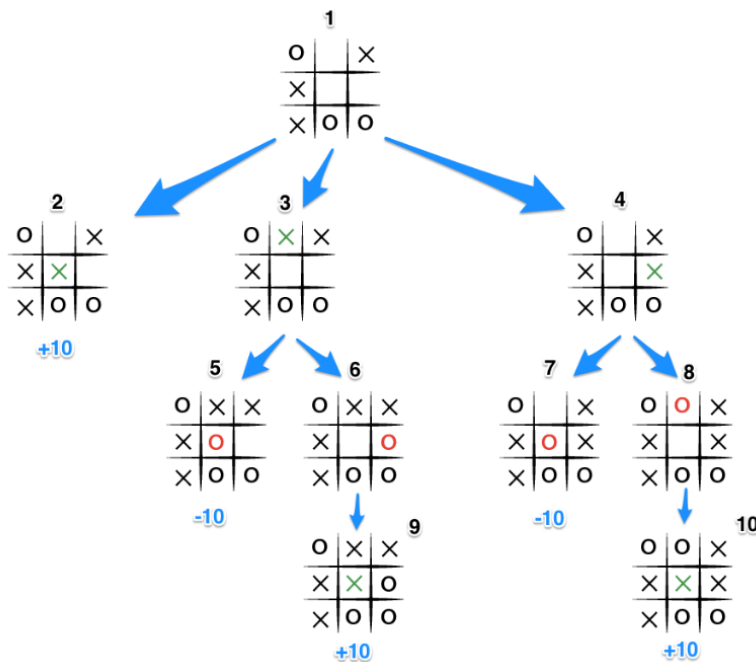
La clave del algoritmo es ir hacia atrás y adelante entre dos jugadores, donde el jugador que tiene el turno desea elegir el movimiento que le da el puntaje máximo. A su vez los puntajes para cada uno de los movimientos disponibles están determinados por el jugador contrario decidiendo cuales de sus movimientos tienen el menor puntaje. Y los puntajes para los movimientos del jugador opositor están determinados por el jugador tratando de maximizar su puntaje y así sucesivamente para todo el árbol de estados.

Una descripción asumiendo X es el jugador de turno sería como:

- Si el juego está terminado, retorna el puntaje desde la perspectiva del jugador X.
- De otra manera obtiene una lista de los nuevos estados para cada posible movimiento.
- Crea una lista de puntajes.
- Para cada uno de esos estados agrega el mínimo resultado de ese estado a la lista de puntajes.
- Si es el turno de X, retorna el máximo puntaje de la lista.
- Si es el turno de O retorna el puntaje mínimo.

El algoritmo es recursivo. El mismo va hacia atrás y adelante entre dos jugadores hasta encontrar el puntaje final.

Ejemplo:



- En el estado 1 es turno del jugador X. Ese jugador genera los estados 2,3 y 4 y llama a minimax para esos estados.
- El estado 2 empuja el puntaje 10 en la lista de puntajes del estado 1, por que el juego está en un estado final.
- Los estados 3 y 4 no son finales, entonces 3 genera 5 y 6 y llama minimax en ellos y el estado 4 genera los estados 7 y 8 y llama minimax en ellos.
- Los estados 6 y 8 generan el único estado posible, los cuales son estados finales, y entonces ambos agregan el puntaje 10 a la lista de movimientos de los estados 3 y 4.
- Como en los estados 3 y 4 es el turno del jugador O, este buscará el mínimo puntaje, y dadas las opciones de 10 y -10 ambos estados 3 y 4 tendrán -10

Finalmente la lista de puntajes para los estados 2, 3 y 4 serán 10 -10 y -10 respectivamente y el estado 1 buscando maximizar el puntaje elegirá el movimiento ganador con puntaje 10, estado 2.

**Artículo original:**

<http://neverstopbuilding.com/minimax>