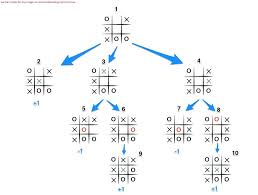
**מינימקס MiniMax**

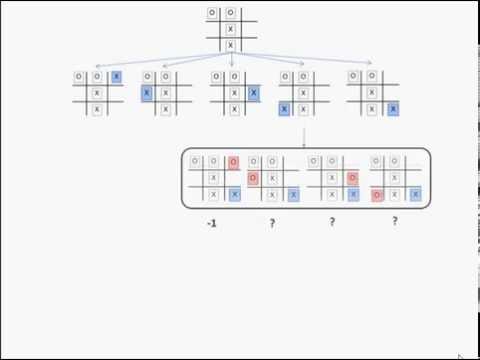
**או**

**איך מחשב מנצח שחמט?**

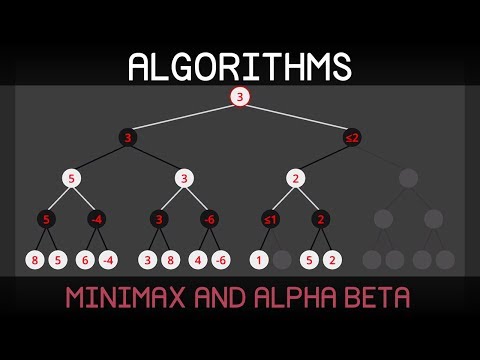
סרטונים:

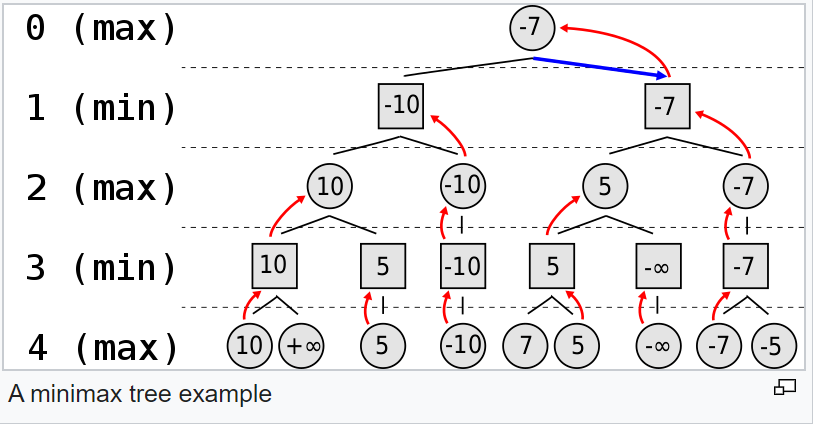
1. [כחול עמוק](https://eureka.org.il/item/22276/%D7%9E%D7%94%D7%95-%D7%94%D7%9E%D7%97%D7%A9%D7%91-%D7%94%D7%A8%D7%90%D7%A9%D7%95%D7%9F-%D7%91%D7%94%D7%99%D7%A1%D7%98%D7%95%D7%A8%D7%99%D7%94-%D7%A9%D7%A0%D7%99%D7%A6%D7%97-%D7%90%D7%AA-%D7%90%D7%9C%D7%95%D7%A3-%D7%94%D7%A2%D7%95%D7%9C%D7%9D-%D7%91%D7%A9%D7%97%D7%9E%D7%98) מנצח את אלוף השחמט

2. [**איקס עיגול**](https://www.youtube.com/watch?v=y7AKtWGOPAE)

[](https://www.youtube.com/embed/0SxPMl5AuiM?feature=oembed)

1. <https://www.youtube.com/watch?v=n8l_WkT8Qzw>

[](https://www.youtube.com/embed/l-hh51ncgDI?feature=oembed)



**מינימקס בקוד**

1. בלולאה ראשית שרצה כל עוד המשחק לא נגמר (אין מנצח ולא תיקו).
2. מבוצעות צעד **שחקן** וצעד **מחשב** לסירוגין כל עוד אין נצחון /תיקו
   1. **צעד שחקן** ,
      1. זיהוי לחיצת מקש – הזזת סמן לבחירת המהלך
      2. קריאה לפעולת **DoPlayerMove**
      3. בדיקת נצחון **CheckWin**
      4. העברת התור למחשב
   2. **צעד מחשב**
      1. חישוב צעד מחשב – **DoPCMove** בעזרת minimax
      2. בדיקת נצחון **CheckWin**
      3. העברת התור לשחקן
3. פעולת **DoPcMove** – פעולה הבודקת את כל האפשרויות למהלך מחשב ומוצאת את המהלך הטוב ביותר (בעזרת minimax) מכל המהלכים אותם היא בדקה ולאחר מכן מבצעת את המהלך

המהלך הטוב ביותר הוא המהלך שיקבל את הציון GetScore הגבוה ביותר. הציון יקבע ע"י פעולה רקורסיבית minimax

**public void DoPCMove()**

**}**

**MinMaxCnt = 0;**

**int move\_i = -1;**

**int max = int.MinValue;**

**Copy Board();**

**for (int i = 0; i < row.Length; i++)**

**}**

**for each possible move**

**{**

**Back to original bard**

**Make move on board**

**int score = MiniMax(true, MAX\_DEEP);**

**if (score >= max)**

**{**

**max = score;**

**move\_i = keep move**

**{**

**}**

**}**

**// make the move\_i with max score.**

**}**

1. פעולת **GetScore** – פעולה שמחזירה "ניקוד" לכל מצב נתון.

ניקוד זה יקבע מהו מהלך טוב יותר.

* 1. מספר חיובי "טוב למחשב"
  2. מספר שלילי – "רע למחשב" (טוב לשחקן)

לדוגמה:

1. נצחון של מחשב יש להחזיר מספר גדול כמו 1000
2. הפסד של מחשב (או נצחון שחקן) יש להחזיר מספר שלילי כמו 1000-
3. ניתן גם לתת ניקוד לפי כמות כלים (כי לא צמיד ניתן להגיע לנצחון בעומק הרקורסיבי.

חישוב פשוט זה לחסר את כמות הכלים של המחשב מכמות הכלים של של השחקן וזה יהיה הציון

1. אם יש משמעות לסוג הכלי (מלכה בדמקה או כלים שונים בשחמט) אז ניצן לתת ניקוד שונה לכל כלי

למשל בשחמט עדיף לאבד רץ ולהרוויח מלכה

**// Return positive or negative**

**// the return value will be high and positive as long as it better to computer**

**// or negetive if it is good to player**

**// in case of win it will return 1000 if loose will return -1000**

**private int GetScore()**

**{**

**return…**

**}**

1. פעולת **MiniMax** - פעולה **רקורסיבית** בעל עומק מוגדר מראש
   1. בודקת את **כל האפשרויות** של מהלך מחשב חוקי
   2. על כל מהלך מחשב שנבדק מופעלת פעולת MiniMax שהפעם בודקת את **כל האפשרויות** של מהלך שחקן חוקי
   3. כך הפעולה קוראת לעצמה - צעד מחשב וצעד שחקן לסירוגין.
   4. על כל מהלך מופעלת פעולת MiniMax פעם לחיזוי מהלך מחשב (Max ) ופעם לחיזוי מהלך שחקן (Min) עד שמגיעים לתנאי עצירה לפי עומק הרקורסיה או נצחון/תיקו
   5. **בתנאי העצירה מוחזר הניקוד – רק אז מתקפלת הרקורסיה ובכל שלב בוחרים את המקסימום (מהלך מחשב) או המינימום (מהלך שחקן) בהתאם**.
   6. אם מהלך מחשב - מכל המהלכים האפשריים ללוח זה יבחר **מקסימום** (טוב למחשב) מכל הציונים
   7. ואם מהלך שחקן – מכל המהלכים האפשריים יבחר **מינימום** (רע למחשב – טוב לשחקן)

**מה בפעולת ה MiniMax?**

1. לולאה העוברת על כל המקומות הפנויים
2. בוחרת מהלך מחשב/ שחקן (תלוי בפרמטר) במקום פנוי כלשהו
3. **לכאורה,** מבצעת את המהלך ומפעילה פעולתMiniMax רקורסיבית למהלך של הצד השני !PlayerTurn
4. **מחזירה חזרה את המהלך** וחוזרת חזרה לתחילת הלולאה לבחירת מהלך אפשרי אחר(במקום להחזיר חזרה ניתן לשמור את הלוח המקורי)
5. בתוך הלולאה יש לבחור את המקס/מינימום (בהתאם לצד המשחק). אם מחשב אז מקס ומחזירה את הניקוד בהתאם

**private int MiniMax(bool playerTurn, int depth)**

**}**

**if (depth ==0 || IsWin() || NoMovesLeft()) // תנאי עצירה**

**return GetScore();**

**int CurrScore = (playerTurn) ? int.MaxValue : int.MinValue;**

**// go over all possibles moves in loop**

**for each possible move call MiniMax**

**{**

**PutMoveOnBoard();**

**score = MiniMax(!playerTurn, depth-1);**

**UndoMove();**

**if (playerTurn && score < currScore) // minimum**

**currScore = score;**

**else if (!PlayerTurn && score> currScore) //maximum**

**currScore = score;**

**}**

**return CurrScore;**

**}**