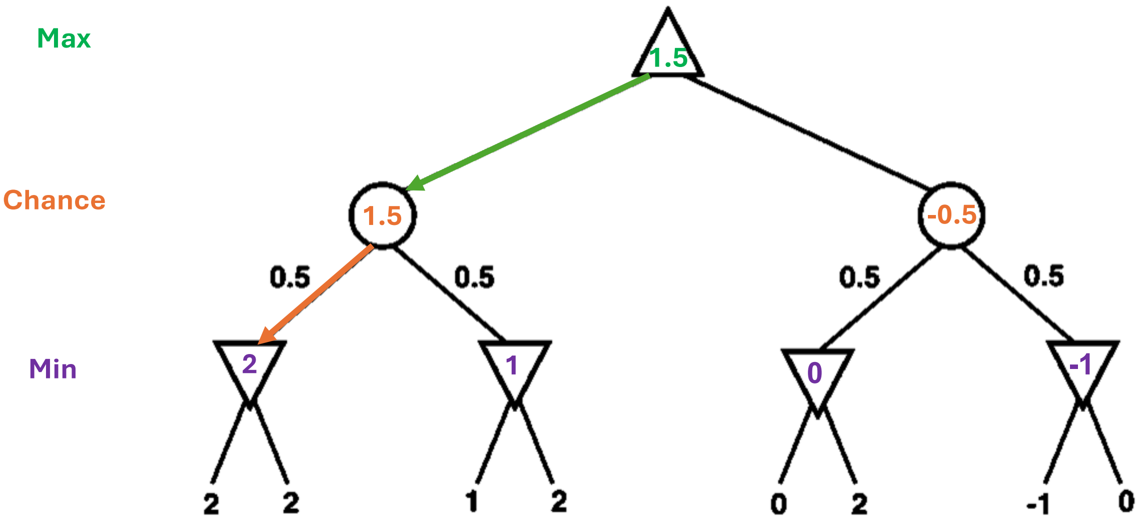
מבוא לבינה מלאכותית 67594, תרגיל 3

ליאורה וסנובטי 211491014 **אופק אבידן 318879574**

.4

4.1



<u>ברמת הMin:</u> בכל קודקוד לקחנו את הערך המינימלי מבין שני הילדים.

<u>ברמת הChance:</u> עבור הקודקוד השמאלי חישבנו את הסיכוי הממושקל וקיבלנו:

$$0.5 * 2 + 0.5 * 1 = 1.5$$

ועבור הקודקוד הימני חישבנו את הסיכוי הממושקל וקיבלנו:

$$0.5 * 0 + 0.5 * -1 = -0.5$$

<u>ברמת הMax:</u> לקחנו את המקסימום מבין $1.5, -0.5$ שהוא 1.5 .

4.2 **בהינתן 6 העלים הראשונים משמאל, נצטרך גם את העלה השביעי והשמיני**. שכן, אנו צריכים לדעת מה יהיה הערך במשולש הימני ביותר (המשולש הרביעי בשכבת המינימום) ואותו נחשב לפי פונקציית מינימום על העלה השביעי והשמיני.
לדוגמא, העלה השביעי והשמיני יכלו להחזיק את הערכים 4 ו-4 בהתאמה, מכאן שהמשולש הימני ביותר בשכבת המינימום היה מחזיק את 4, העיגול השני משמאל בשכבת ה-Chance היה מחזיק את הערך 2, ומכאן שהיינו משנים את המסלול שלנו, שלא כמו באיור המקורי. כלומר ערכי העלים השביעי והשמיני הם מידע הכרחי כדי לבחור את מסלול העץ.
לעומת זאת, **לאחר שאנחנו כבר יודעים את 7 העלים הראשונים משמאל**, **לא נצטרך את העלה השמיני.** שכן, בהינתן העלה השביעי, אנו יודעים שהמשולש הרביעי משמאל בשכבת המינימום, מחזיק לכל היותר את הערך -1 (כי העלה השביעי מחזיק את הערך -1 ומבצעים פונקציית מינימום). מכאן, שאנחנו נקבל ערך שקטן (או שווה) מ-0.5, ולכן בהכרח סוכן המקסימום ילך שמאלה בעץ (שכן אנו יודעים כבר את ערכו, שהוא 1.5).

4.3 בהינתן שכל עלה נמצא בטווח $[-2,2]$, וכן בהינתן 2 העלים השמאליים של העץ הנתון, נחשב את הטווח של ה-node הראשון בשכבת ה-Chance. הערך הנמוך ביותר שהמשולש השני משמאל בשכבת ה-Min יכול להגיע אליו, הוא -2, שכן הערך הנמוך ביותר שכל עלה יכול להגיע אליו הוא -2 ולאחר מכן אנו משתמשים בפונקציית Min על העלים.
מכך ש-$-1 = -2 \cdot 0.5$, אז הערך הנמוך ביותר שנגיע אליו ב-node הראשון משמאל בשכבת ה-Chance הוא $-1 + 1 = 0$, כסכום שני המשולשים הראשונים בשכבת ה-Min.

מנגד, הערך הגבוה ביותר שהמשולש השני משמאל בשכבת ה-Min יכול להגיע אליו, הוא 2, שכן הערך הגבוה ביותר שכל עלה יכול להגיע אליו הוא 2 ולאחר מכן אנו משתמשים בפונקציית Min על העלים.
לכן, ומכך ש-$2\cdot 0.5=1$, אז הערך הנמוך ביותר שנגיע אליו ב-node הראשון משמאל בשכבת ה-Chance הוא $1+1=2$, כסכום שני המשולשים הראשונים בשכבת ה-Min.
בסה"כ נקבל שהערך הנמוך ביותר הוא 0 והערך הגבוה ביותר הוא 2, כלומר בסה"כ נקבל את הטווח $[\mathbf{0},\mathbf{2}]$.

4.4 בהינתן ההנחות של שאלה 3 (כלומר, אנו יודעים את שני העלים הראשונים, ואנו יודעים ששאר העלים בטווח $[-2{,}2]$), מספיק יהיה לנו להבין את **5 העלים הראשונים** כדי לדעת את הבחירה של סוכן ה-*Max*.
נשים לב שאם אנו יודעים רק את **ערכי שני העלים הראשונים**, אנו רק נדע שה- *node* הראשון בשכבת ה-*Chance* הוא בטווח $[0{,}2]$, והשני בשכבת ה-*Chance* הוא בטווח $[-2{,}2]$, ולכן סוכן המקסימום לא יידע באיזה *node* לבחור.
אם נדע את **ערכי שלושת העלים הראשונים**, נדע כל מה שידענו מהשניים, ובנוסף, אנו נדע שהמשולש הראשון (בשכבת ה-*Min*) מכיל את הערך 2, והמשולש השני מכיל ערך אשר קטן או שווה ל-1. לכן ה-*node* הראשון בשכבת ה-*Chance* יהיה בטווח $[0{,}1.5]$, שכן הערך הקטן ביותר שיכול להיות במשולש השני לא השתנה ממקודם, אבל הערך הגדול ביותר שיכול להיות במשולש השני יהיה 1, כלומר בסה"כ $0.5\cdot 2+0.5\cdot 1=1.5$. הטווח של ה-*node* השני יישאר $[-2{,}2]$, ושוב סוכן המקסימום לא יידע באיזה *node* לבחור (ייתכן שה-*node* השני יכיל 2 ואז יצטרך ללכת ימינה, ויתכן שיכיל -2 ואז יצטרך ללכת שמאלה).
אם נדע את **ערכי ארבעת העלים הראשונים**, נדע את כל מה שידענו מהשלושה, ובנוסף, נדע שהמשולש השני בשכבת ה-*Min* מכיל את הערך 1, ומכך שה-*node* הראשון בשכבת ה-*Chance* מכיל את הערך 1.5. לא התווסף שום מידע לגבי החלק הימני של העץ, ולכן ה-*node* השני בשכבת ה-*Chance* יישאר בטווח $[-2{,}2]$. כלומר, שוב סוכן המקסימום לא יידע באיזה *node* לבחור (ייתכן שה-*node* השני יכיל 2 ואז יצטרך ללכת ימינה, ויתכן שיכיל -2 ואז יצטרך ללכת שמאלה).
אם נדע את **ערכי חמשת העלים הראשונים**, נדע את כל מה שידענו מהרביעייה, ובנוסף, נדע שהמשולש השלישי בשכבת המינימום מכיל ערך שקטן או שווה ל-0. כלומר, הערך שהמשולש השלישי בשכבת המינימום יוסיף ל-node השני בשכבת ה-Chance הוא לכל היותר $0\cdot 0.5=0$. לכן, כעת הטווח של ה-*node* יהיה $[-2{,}1]$, שכן הטווח של המשולש הרביעי בשכבת המינימום לא השתנה, והטווח של המשולש השלישי הצטמצם ל-$[-2{,}0]$. לכן, כעת **סוכן המקסימום יידע לבחור ב-*node* השמאלי בשכבת ה-*Chance*,** שכן, הוא יודע שהקודקוד הזה מכיל את הערך 1.5, ואילו הקודקוד הימני יכול להכיל ערך שהוא לכל היותר 1. מפונקציית מקסימום, הבחירה ברורה ונסיים.

.6

6.1.1 תיאורטית, האלגוריתם **Expectimax** הכי מתאים מבין שלושת האלגוריתמים למשחק 2048.
כפי שראינו בתרגול, אלגוריתם <u>הMinimax</u> מיועד למשחקי zero-sum, כאשר כל שחקן שואף למקסם את הניקוד שלו תוך מזעור הניקוד של היריב. האלגוריתם מניח שהיריב משחק בצורה אופטימלית. עם זאת, למשחק 2048 אין יריב במובן שדיברנו עליו בכיתה. במקום זאת, ה"יריב" של המשחק הוא המשחק עצמו- שמפיק את האריחים האקראיים ומכניס אלמנטים סטוכסטיים למשחק. כלומר, בסך הכל, אלגוריתם זה אינו המיטבי לסביבה האקראית הזו.
כפי שלמדנו בכיתה, אלגוריתם ה<u>Alpha-Beta-pruning</u> מנסה למטב את אלגוריתם הMinimax ע"י הפחתת מספר הצמתים המוארכים בעץ המשחק, ע"י גזימת הענפים הלא משפיעים על ההחלטה הסופית. נשים לב, שההנחה על היריב שבוצעה באלגוריתם הMinimax עדיין מתקיימת גם באלגוריתם זה ולכן גם אלגוריתם זה אינו המיטבי עבור

המשחק 2048. לעומת זאת, אלגוריתם הExpectimax הוא וריאציה של אלגוריתם הMinimax המיועד למשחקים עם מאורעות אקראיים. כלומר, באלגוריתם זה, ההנחה היא שהיריב מכניס אקראיות למשחק. כפי שציינו, במשחק 2048, היריב אכן מכניס אקראיות למשחק ע"י הפקת אריחים אקראיים. אלגוריתם הExpectimax מחשב ממוצע של כל התוצאות האפשריות של המאורעות האקראיים ובוחר לפי זה את הצעד המתאים. לכן אלגוריתם זה הוא המתאים ביותר למשחק שלנו.

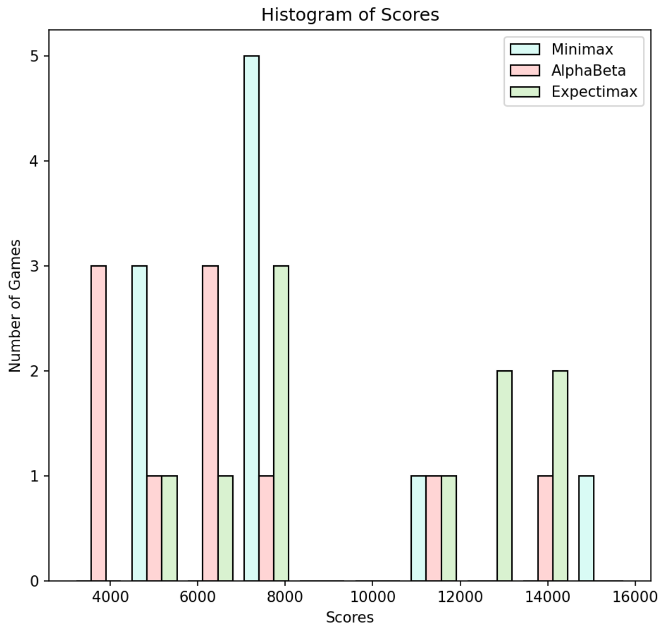
6.1.2 נציג טבלה של ההרצות של כל אחד מהאלגוריתמים עבור depth=2 ו 10 משחקים:

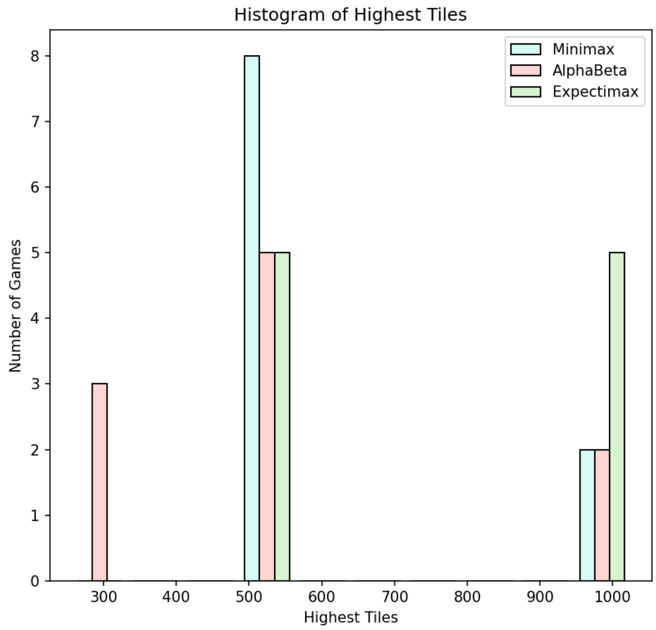
Game number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Minimax score	5424	7236	7124	7128	15864	5552	6944	11528	7056	5436
Minimax highest tile	512	512	512	512	1024	512	512	1024	512	512
AlphaBeta score	5312	3096	3448	3252	6568	7360	13632	6124	11580	6864
AlphaBeta highest tile	512	256	256	256	512	512	1024	512	1024	512
Expectimax score	14544	5252	12184	6716	14460	11104	12356	7200	7092	7124
Expectimax highest tile	1024	512	1024	512	1024	1024	1024	512	512	512

אמפירית, ניתן לראות שאלגוריתם ה **Expectimax** הוא הכי יעיל מבין השלושה.
זאת מאחר שניתן לראות בטבלה, שאלגוריתם זה מצליח לקבל 1024 בחצי מההרצות, לעומת שני האלגוריתמים האחרים שהצליחו להגיע ל1024 רק פעמיים.

6.1.3 ניתן לראות שהתוצאות האמפיריות תואמות למה שציפינו תיאורטית. כפי שציפינו תיאורטית, אלגוריתם ה**Expectimax** הוא הכי יעיל מבין השלושה. בטבלה ב6.2 ניתן לראות שאלגוריתם ה **Expectimax** הצליח לקבל 1024 בחצי מההרצות, לעומת שני האלגוריתמים האחרים, שהצליחו להגיע ל1024 רק פעמיים. מכאן ניתן להסיק שאלגוריתם ה**Expectimax** יותר יעיל.

נייצג את התוצאות בהיסטוגרמה:
ניתן לראות בגרף השמאלי, כי אלגוריתם ה-Expectimax הגיע מספר פעמים (4) לתוצאות גבוהות יחסית (מעל 12,000), ואילו שני האלגוריתמים האחרים, Minimax ו-AlphaBeta הגיעו רק פעם אחת לתוצאה מעל 12,000.
בגרף הימני, ניתן לראות שבחצי מהמשחקים (5) אלגוריתם ה-Expectimax הגיע ל- Max Tile ששווה ל-1024. זאת בניגוד לשני האלגוריתמים האחרים, Minimax ו-AlphaBeta, אשר הגיעו ל-Max Tile ששווה ל-1024 רק פעמיים לכל אלגוריתם.





6.1.4 נחשב את סטיית התקן של הScore בשני האלגוריתמים (Expectimax and Minimax):

$$\sigma = \sqrt{\frac{1}{10}\sum_{i=1}^{10}(score_i - \mu)^2}$$

	ממוצע	סטיית תקן
Expectimax	9803.2	3310.402
Minimax	7929.2	3123.450

נחשב את סטיית התקן של הhighest tile בשני האלגוריתמים (Expectimax and Minimax):

$$\sigma = \sqrt{\frac{1}{10}\sum_{i=1}^{10}(score_i - \mu)^2}$$

	ממוצע	סטיית תקן
Expectimax	768.0	256.0
Minimax	614.4	204.8

הסבר אינטואיטיבי להבדלים בתוצאות:
Minimax הוא אלגוריתם שממזער את ההפסד האפשרי בתרחיש הגרוע ביותר, בהנחה שהיריב משחק בצורה אופטימלית. הוא מעריך את כל המהלכים האפשריים ואת התוצאות שלהם כדי לבחור את המהלך הפחות גרוע. הגישה הדטרמיניסטית שלו, שבה האלגוריתם תמיד עוקב אחר אסטרטגיה צפויה כדי למקסם את הציון תוך מזעור הפסדים פוטנציאליים גורמת לעקביות בפעולתו. כתוצאה מכך, ה-Score וה-Highest tile שהושגו על ידי Minimax מראים פחות שונות, המתבטאת בסטיות התקן הנמוכות שלו.
גם ב Score וגם בHighest tile, סטיות התקן שקיבלנו קטנות מסטיות התקן שקיבלנו באלגוריתם הExpectimax. בScore קיבלנו סטיית תקן של 3123.450 , בHighest tile סטיית התקן היא 204.8.
Expectimax, לעומת זאת, נוקט בגישה הסתברותית יותר. הוא מעריך את התוצאות הצפויות על ידי התחשבות בהסתברות של כל מקרה אפשרי. גישה זו מאפשרת ל- Expectimax

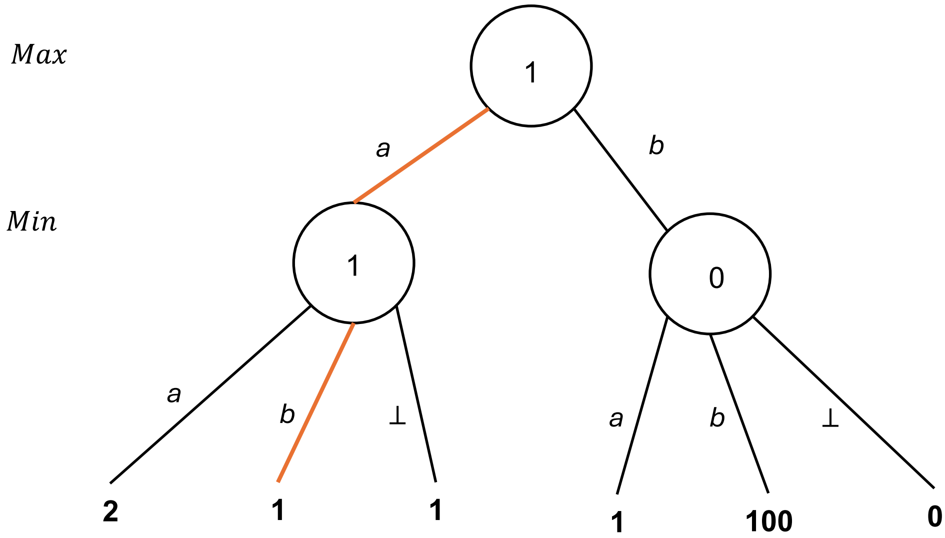
להתנהג בצורה אקראית יותר בפעולתו. האלגוריתם יכול לפעמים להשיג ציונים גבוהים מאוד אבל גם מסתכן בציונים נמוכים יותר אם האלמנטים האקראיים של המשחק לא יתפתחו בצורה חיובית. לכן, אלגוריתם זה מראה עוד יותר שונות בביצועים שלו בכל משחק.
ואכן ניתן לראות שסטיית התקן גם בscore וגם בhighest tile יותר גבוהה מאלה של אלגוריתם הminimax. מבחינת Score, ל- Expectimax יש סטיית תקן של 3310.402, גבוה מ-Minimax. מבחינת ה-Highest tile, ל- Expectimax יש סטיית תקן של 256, גבוה מ-Minimax.
ניתן לראות זאת גם בהיסטוגרמות שצירפנו ב6.3:
בהיסטוגרמת הScore אלגוריתם ה-Minimax הגיע לאותו Score ב5 מתוך 10 המשחקים (העמודה באיזור ה7000), כלומר שונות נמוכה, בעוד שאלגוריתם ה-Expectimax הגיע לאותו Score לכל היותר ב-3 מתוך 10 המשחקים (העמודה באיזור ה-8000). כמו כן, בהיסטוגרמה של הHighest tile , אלגוריתם ה-Minimax הגיע לאותו Highest tile ב8 מתוך 10 המשחקים (העמודה ב-512), כלומר שונות נמוכה, בעוד שאלגוריתם ה-Expectimax הגיע לאותו Highest tile לכל היותר ב-5 מתוך 10 המשחקים (העמודות ב-1024 וכן העמודה ב-512).

6.2.1 נניח כעת שמשחקים במשחק האלטרנטיבי- "2048 בום" בו לאחר כל צעד של השחקן יש סיכוי של 1 לביליון ל"בום", כלומר למצב בו המשחק מסתיים והשחקן מסיים עם score 0.
<u>השינוי לא ישפיע רבות על אלגוריתם הExpectimax:</u>
ניזכר כי אלגוריתם זה נוקט גישה הסתברותית. בכל צעד, ההסתברות לכל מקרה אפשרי מתפלגת באופן אחיד למעט ההסתברות למצב בו קורה "בום". האלגוריתם לוקח בחשבון את ההסתברויות לכל המקרים האפשריים. לכן, במשחק האלטרנטיבי 2048 בום, נצטרך בסך הכל להוסיף לחישוב את ההסתברות $\frac{1}{1,000,000,000}$ (הסתברות מאוד זניחה)- ההסתברות שבצעד הבא יקרה "בום" והמשחק יסתיים. הסתברות זו זניחה ולכן לא תשפיע מדי על הביצועים של אלגוריתם זה.

לעומת זאת, <u>השינוי ישפיע על אלגוריתם הMinimax:</u>
ניזכר כי אלגוריתם זה נוקט גישה יותר דטרמיניסטית ועקבית. נבחין כי בכל node בשכבת ה-minimum הסוכן ייבחר את ה-action הטוב ביותר מבחינתו, כלומר "בום". זאת בשל הגדרתו של האלגוריתם, לפיו כל סוכן מבצע את המהלך האופטימלי לכל node. בנוסף, סוכן ה-maximum מניח שסוכן ה-minimum מבצע את המהלך האופטימלי עבורו (עבור סוכן ה-minimum). כלומר סוכן ה-maximum חושב שסוכן ה-minimum בחר "בום" לכל node.
לכן, כל קודקוד בשכבת הminimum יהיה בעל הערך 0, ומכך שכל קודקוד בשכבת המקסימום גם הוא יהיה בעל הערך 0 ויבחר בחירה אקראית בין כל הactions האפשריים.

6.2.2 ניצור משחק חדש
<u>כללי המשחק:</u>
- משחק עבור שחקן יחיד, אשר מתחיל עם ניקוד 0.
- לאחר כל מהלך, יש סיכוי של 99% שהמשחק יסתיים עם הניקוד הנוכחי.
- בכל מהלך, אפשר לבחור a או b. a מוסיף לשחקן נקודה אחת בדיוק. b מוסיף 0 נקודות.
- אם בוחרים b בשני מהלכים ברצף, זה מוסיף לשחקן 100 נקודות.

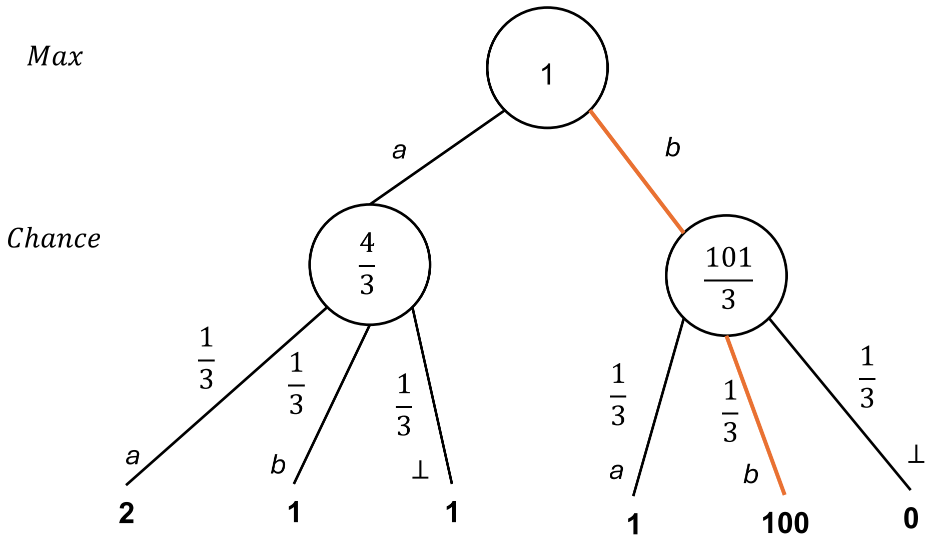
כאשר אנחנו מתייחסים לתור יחיד, אלגוריתם ה-minimax ייבחר את a, שכן לפי הגרף מתקיים:



ב-node השמאלי של שכבת ה-minimum, סוכן ה-minimum ייבחר ב-b (או שהמשחק ייעצר), ולכן ערך ה-node השמאלי יהיה 1. ב-node הימני, סוכן ה-minimum ייבחר לעצור את המשחק ולכן ערך ה-node הימני יהיה 0.

מכאן שסוכן ה-maximum יבחר ללכת שמאלה, כלומר כשמדובר בתור יחיד אלגוריתם ה-minimax ייבחר ב-a, ולאחר מכן המשחק כמעט-בוודאות יסתיים (בהסתברות של 99%), עם score=1.

לעומת זאת, אלגוריתם ה-not-exactly-expectimax ייבחר ב-b בתור יחיד, שכן לפי הגרף מתקיים:



ב-node השמאלי של שכבת ה-Chance, הערך הוא $\frac{4}{3}$ שכן מתקיים $\frac{1}{3}\cdot 2+\frac{1}{3}\cdot 1+\frac{1}{3}\cdot 1=\frac{4}{3}$, כאשר ערך כל אחד מהעלים הוא הסכום שיתקבל מהצעדים הנבחרים.

ב-node הימני של שכבת ה-Chance, הערך הוא $\frac{101}{3}$ שכן מתקיים $\frac{1}{3}\cdot 1+\frac{1}{3}\cdot 100+\frac{1}{3}\cdot 0=\frac{101}{3}$. כאשר ערך כל אחד מהעלים הוא הסכום איתו נסיים את המשחק.

לכן, מאופן פעולתו של אלגוריתם ה-not-exactly-expectimax, הסוכן ייבחר ללכת ל-node הימני, כלומר ב-b.
עם זאת, יש לזכור כי לאחר התור הראשון בו הסוכן בוחר את b, ישנה הסתברות של 99% שהמשחק יסתיים, כלומר 99% שהמשחק יסתיים בscore=0.

בסה"כ אנו רואים כי בהסתברות של 99% המשחק יסתיים בתוצאה 0 אם נבחר באלגוריתם ה-not-exactly-expectimax, וכי בהסתברות של 99% המשחק יסתיים בתוצאה 1 אם נבחר באלגוריתם הminimax. מכאן שאלגוריתם הminimax נותן ביצועים יותר טובים מאלגוריתם הexpectimax כנדרש.

נראה כי השחקן שמשחק לפי אלגוריתם הMinimax נותן ביצועים יותר טובים מהשחקן שמשחק לפי אלגוריתם הExpectimax בתוחלת:

נגדיר מ"מ חדש $X_{Minimax}$ שהוא תוצאת המשחק של שחקן הminimax לאחר שני תורות.
נגדיר מ"מ חדש $X_{Expectimax}$ שהוא תוצאת המשחק של שחקן הexpectimax לאחר שני תורות.

תוחלת $X_{Minimax}$:

$$\mathbb{E}(X_{Minimax}) = 1 + \frac{1}{100}$$

שכן בצעד הראשון הוא יבחר את a ויקבל נקודה, בצעד השני (נגיע לתור זה בהסתברות של $\frac{1}{100}$) יבחר שוב את a ויקבל נקודה.
תוחלת $X_{Expectimax}$:

$$\mathbb{E}\left(X_{Expectimax}\right) = 0 + \frac{1}{100} * 100 = 1$$

שכן בצעד הראשון נבחר את *b* ונקבל 0 נקודות ובצעד השני (נגיע לתור זה בהסתברות של $\frac{1}{100}$) נבחר שוב את *b* ונקבל 100 נקודות.

מכאן ניתן לראות שמתקיים: $\mathbb{E}\left(X_{Expectimax}\right) < \mathbb{E}(X_{Minimax})$. כלומר שחקן הminimax נותן ביצועים טובים יותר משחקן הexpectimax בתוחלת כנדרש.